

---

# Mètodes per a la resolució d'equacions no lineals amb l'operador diferències dividides

---

UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC) - BARCELONATECH

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)

Grau en Enginyeria Informàtica  
Especialitat Computació

Departament de Matemàtica Aplicada II

*Autor:*

Sandra Corella Pérez

*Director:*

Miquel Grau Sàncnez

Data de defensa:

1 de juliol de 2015



## Resum

Aquest treball titulat *Mètodes per a la resolució d'equacions no lineals amb l'operador diferències dividides* tracta sobre la resolució d'equacions no lineals mitjançant mètodes iteratius amb la utilització de l'operador diferències dividides i multiprecisió.

A partir dels mètodes iteratius clàssics d'un pas com són els de la Secant, Kurxatov i Steffensen es construeixen esquemes iteratius de dos passos emprant la pseudo-composició o la congelació de l'operador en els dos passos d'iteració. El cost computacional i l'ordre local de convergència dels mètodes determina una classificació d'aquests, segons l'eficiència. Es calcula també el radi de convergència de l'algorisme iteratiu per tal d'establir la mesura de l'interval on el mètode convergeix des d'un punt de vista teòric i numèric. Finalment, es presenta un estudi comparatiu entre la resolució mitjançant una aritmètica fixa (AF) i una aritmètica adaptativa (AA) com també una comparativa entre tres llibreries de multiprecisió.

## Resumen

Este trabajo titulado *Métodos para la resolución de ecuaciones no lineales con el operador diferencias divididas* trata sobre la resolución de ecuaciones no lineales a partir de métodos iterativos con la utilización del operador de diferencias divididas y la multiprecisión.

A partir de los métodos iterativos clásicos de un paso como son la Secante, Kurxatov y Steffensen se construyen esquemas iterativos de dos pasos utilizando la pseudo-composición o la congelación del operador en los dos pasos de iteración. El coste computacional y el orden local de convergencia de los métodos determina una clasificación de éstos, según la eficiencia. Se calcula también el radio de convergencia del algoritmo iterativo para establecer la medida del intervalo donde el método converge desde un punto de vista teórico y numérico. Finalmente, se presenta un estudio comparativo entre la resolución mediante una aritmética fija (AF) y una aritmética adaptativa (AA), así como una comparativa entre tres librerías de multiprecisión.

## **Abstract**

This work titled *Methods for resolution of nonlinear equations with the divided difference operator* is about the resolution of nonlinear equations using iterative methods with the divided differences operator and multiprecision.

From the classical one step iterative methods such as Secant, Kurchatov and Steffensen two steps iterative schemes are build using the pseudo-composition and frozen operator in both steps of the iteration. The computational cost and the local order of convergence define a classification of the methods according their efficiency. The convergence radius is also calculated to establish the size of the interval where the method converges. Finally a comparative study between the resolution using fixed arithmetic and adaptative arithmetic as well as a comparative between three different multiprecision libraries are presented.



# Índex

<b>1</b>	<b>Definició del projecte</b>	<b>1</b>
1.1	Context i motivació . . . . .	1
1.2	Estat de l'art . . . . .	2
1.3	Objectius . . . . .	3
<b>2</b>	<b>Introducció</b>	<b>5</b>
2.1	Introducció . . . . .	5
2.2	Mètodes iteratius que utilitzen l'operador de diferències dividides . . . . .	6
2.3	Definicions de l'ordre de convergència . . . . .	6
2.4	Eficiència computacional . . . . .	8
2.5	Ordres Computacionals de convergència . . . . .	10
2.5.1	COC . . . . .	10
2.5.2	ACOC . . . . .	10
2.5.3	ECOC . . . . .	10
2.5.4	PCOC . . . . .	11
2.5.5	Definicions . . . . .	11
2.6	Criteris de tall . . . . .	12
2.7	Aritmètiques . . . . .	13
<b>3</b>	<b>Mètodes Iteratius</b>	<b>15</b>
3.1	Mètodes iteratius . . . . .	15
3.1.1	Mètodes utilitzant pseudo-composició . . . . .	15
3.1.2	Mètodes amb operador congelat . . . . .	16
3.1.3	Mètodes pseudo-composats amb si mateixos . . . . .	17
3.1.4	Altres mètodes . . . . .	17
3.2	Estudi de l'ordre de convergència. Desenvolupaments . . . . .	18
3.3	Equacions dels errors i ordres locals de convergència . . . . .	19
3.3.1	Pseudo-composats . . . . .	20
3.3.2	Congelats . . . . .	21

3.3.3	Pseudo-composats amb si mateixos . . . . .	23
3.3.4	Altres mètodes . . . . .	23
<b>4</b>	<b>Eficiències</b>	<b>25</b>
4.1	Estudi de les eficiències . . . . .	25
4.2	Radi de convergència. Estudi Local . . . . .	30
4.2.1	Tècnica genèrica . . . . .	30
4.2.2	Mètode de la Secant . . . . .	31
4.2.3	Mètode de Shakno . . . . .	32
4.2.4	Mètode de Traub . . . . .	32
4.2.5	Mètode de Kurxatov . . . . .	33
4.2.6	Mètode congelat de Steffensen . . . . .	34
4.3	Temps d'execució d'operacions bàsiques i funcions elementals . . . . .	35
<b>5</b>	<b>Resultats numèrics</b>	<b>37</b>
5.1	Resultats amb Aritmètica Fixa . . . . .	39
5.2	Resultats amb aritmètica adaptativa . . . . .	45
5.3	Comparativa entre AF i AA . . . . .	47
5.4	Aproximacions inicials i Número d'iteracions . . . . .	49
5.5	Càlcul del radi de convergència a partir de K, L i M . . . . .	54
5.6	Comparativa entre MPFR, ARPREC i BigFloat . . . . .	56
5.7	Conclusions dels resultats numèrics . . . . .	59
<b>6</b>	<b>Implementació</b>	<b>61</b>
6.1	Disseny . . . . .	61
6.2	Algorisme genèric . . . . .	64
<b>7</b>	<b>Planificació del projecte</b>	<b>67</b>
7.1	Planificació temporal . . . . .	67
7.1.1	Tasques . . . . .	68
7.1.2	Requeriments i anàlisi de dependències . . . . .	69
7.2	Planificació econòmica . . . . .	72
7.2.1	Identificació de costos . . . . .	72
7.2.2	Estimació de costos . . . . .	72
7.3	Sostenibilitat i compromís social . . . . .	73
<b>8</b>	<b>Conclusions</b>	<b>75</b>
	<b>Bibliografia</b>	<b>76</b>
	<b>Annex</b>	<b>80</b>



<b>A</b>	<b>Taules complementàries dels resultats numèrics</b>	<b>81</b>
A.1	Aritmètica Fixa (AF) . . . . .	81
A.2	Aritmètica Adaptativa (AA) . . . . .	81
A.3	Comparativa entre AA i AF . . . . .	81
<b>B</b>	<b>Codi</b>	<b>83</b>
B.1	Sistema de resolució d'equacions no lineals . . . . .	83
B.1.1	MPFR . . . . .	83
B.1.2	ARPREC . . . . .	84
B.1.3	BigFloat . . . . .	84
B.2	Eficiències . . . . .	84
B.3	Radis . . . . .	84
B.4	Aproximacions inicials i número d'iteracions . . . . .	84



# Capítol 1

## Definició del projecte

### 1.1 Context i motivació

La matemàtica aplicada requereix, en general, resoldre models matemàtics en els quals, en molts casos, apareix la resolució d'una equació o un sistema d'equacions no lineals. A causa d'aquesta necessitat, l'interès per la construcció de mètodes iteratius d'ordre superior per resoldre equacions ha retornat a aquesta branca de l'anàlisi numèrica que és una via d'investigació molt destacada.

Els problemes que es resolen en els camps d'investigació i recerca solen estar condicionats per un gran nombre de variables, i la resolució de les equacions no lineals a partir de mètodes directes té un cost computacional no assolible, en el cas que tingui solució analítica. Molt sovint és necessari fer servir un mètode iteratiu per a la seva resolució.

Són exemple d'aquesta situació: la modelització del clima, el càlcul de les òrbites planetàries, les simulacions de supernoves i de fluids com també la resolució d'alguns aspectes de la teoria quàntica [5].

Els mètodes iteratius tracten de resoldre una equació mitjançant aproximacions successives que es van acostant a la solució a partir d'una aproximació inicial. En augmentar l'ordre de convergència del mètode cal fer menys iteracions per arribar a la solució, entenent com a solució l'aproximació amb un nombre predeterminat de xifres significatives.

L'operador en diferències dividides aproxima la primera derivada en un punt interior. Aquest operador fa possible la resolució de l'equació sense el costós càlcul de la derivada de la funció, que en molts casos no es pot calcular.

A partir dels mètodes clàssics de la Secant, Kurxatov i Steffensen, que són mètodes iteratius d'un pas,

es construeixen esquemes iteratius de dos passos a partir de la pseudo-composició o la congelació de l'operador en els dos passos d'iteració. S'introdueixen els mètodes de Traub i Shakno ja que també són algorismes iteratius de dos passos.

A partir del cost computacional i l'ordre local de convergència del mètode es poden classificar els mètodes iteratius en termes d'eficiència per veure quin seria més eficient per resoldre les equacions amb el programari escollit.

Es calcula també el radi de convergència del mètode iteratiu per tal d'establir la mesura de l'interval on el mètode convergeix des d'un punt de vista teòric i numèric. Els radis de convergència entre els diferents mètodes iteratius poden variar i també serà un factor a tenir en compte.

Un gran nombre d'aplicacions de computació científica requereix ser resolt amb precisions superiors a doble o quàdruple precisió. Per aquest motiu les aritmètiques d'alta precisió, multiprecisió, han esdevingut un element indispensable per a l'entorn de computació científica a gran escala.

Amb aquesta aritmètica amb multiprecisió s'introdueix l'aritmètica adaptativa i es compara amb la clàssica veient la millora en temps d'execució per a la resolució d'equacions no lineals.

## 1.2 Estat de l'art

Els tres esquemes iteratius clàssics sense derivades són el mètode de la Secant [4], el mètode de Kurxatov [21, 29] i el mètode d'Steffensen [2, 3, 30].

Es diu que un mètode de dos passos és congelat quan l'operador es repeteix [14]. La nomenclatura que s'utilitza s'ha introduït en [27].

A partir dels mètodes clàssics, amb memòria i sense memòria, es construeixen mètodes de dos passos utilitzant pseudo-composició i congelació de l'operador [9]. Una altra manera de construir mètodes amb l'operador congelat i amb memòria es presenten a Shakno [29] i Traub [31].

El concepte d'ordre de convergència tal com es treballa actualment l'ha introduït Traub (1964, [33]). Una altra definició ja havia estat presentada per Wall (1956, [34]). Aquesta última es fa servir en aquest treball quan es calcula l'ordre de convergència numèricament. Tot l'estudi dels errors, per tal de trobar els ordres de convergència dels mètodes estudiats, es farà a partir de les equacions en diferències de l'error. En el cas dels mètodes iteratius amb memòria es farà servir l'equació indicial de l'error [32].

Els paràmetres d'eficiència que es fan servir són l'índex d'eficiència computacional (*CEI*, *Computational efficiency index*) [15] i el factor temps (*TF*, *Time Factor*). Altres autors com Traub i Ostrowski fan servir casos particulars de *CEI* anomenats índex d'eficiència [25] (*EI*, *Efficiency Index*) i eficiència computacional [33] (*CE*, *Computational Efficiency*), respectivament. Les tècniques de classificació per eficiència es poden trobar a [16].

Com a paràmetres de càlcul aproximat de l'ordre es tenen:

- L'ordre de convergència computacional (*COC*, *Computational Order of Convergence*) [35].
- L'ordre de convergència computacional aproximat (*ACOC*, *Approximated Computational Order of Convergence*) [22].
- L'ordre de convergència computacional extrapolat (*ECOC*, *Extrapolated Computational Order of Convergence*) [22], que fa servir l'extrapolació d'Aitken [1].
- L'ordre de convergència computacional de Petković (2011, [26]) (*PCOC*, *Petković Computational Order of Convergence*).

Els ordres computacionals de convergència local (*CLOC*, *ACLOC*, *ECLOC* i *PCLOC*) van ser introduïts per Grau-Sánchez et al. (2012, [18]); aquests paràmetres són els que es fan servir en aquest treball.

Actualment se segueixen estudiant mètodes d'un i dos passos per a la resolució de sistemes d'equacions no lineals [8, 10, 19, 17], com també se segueix desenvolupant el càlcul computacional de l'operador en diferències dividides [11] i perfeccionant els càlculs d'ordre i d'eficiència [13, 20].

La resolució d'equacions no lineals requereix en molts casos una aritmètica amb multiprecisió. Amb aquesta finalitat s'ha utilitzat la llibreria MPFR [12]. Un avantatge d'aquesta llibreria, a més d'acceptar els càlculs en multiprecisió, resideix que accepta el canvi d'aritmètica, és a dir canvis de longitud de la mantissa, adequant l'aritmètica a la precisió desitjada en cada iteració amb molt poc cost computacional. Altres llibreries com són ARPREC i BigFloat s'han fet servir per mesurar l'*speedup* dels temps d'execució respecte MPFR.

### 1.3 Objectius

L'objectiu principal d'aquest projecte és fer una investigació dels esquemes iteratius per a la resolució d'equacions no lineals que fan ús de l'operador de diferències dividides.

El seguit d'objectius que inclou són:

1. Construcció de nous esquemes iteratius de dos passos a partir de mètodes de resolució: Secant, Kurxatov i Steffensen.
2. Consideració dels mètodes de Shanno i Traub com a mètodes coneguts de dos passos amb un ordre de convergència alt.
3. Càlcul i anàlisi de l'ordre local de convergència.
4. La classificació dels algorismes iteratius respecte del cost computacional i l'ordre de convergència local.
5. L'estudi teòric i pràctic del radi de convergència dels mètodes més eficients.
6. Consideració de diferents paràmetres per al càlcul dels errors i dels ordres locals de convergència.
7. Utilització dels mètodes més eficients per un conjunt de funcions mostra, per tal d'obtenir la confirmació de tots els resultats teòrics.
8. Utilització d'una aritmètica de multiprecisió de 4096 dígit per a la resolució d'equacions i sistemes no lineals.
9. Minimització dels temps d'execució dels mètodes amb una aritmètica fixa i adaptativa.
10. Desenvolupament dels algorismes iteratius amb el programari MPFR, ARPREC i BigFloat per tal de mesurar-ne els temps d'execució de cadascun.

## Capítol 2

# Introducció

### 2.1 Introducció

Sigui  $f : J \subset \mathbb{R} \rightarrow \mathbb{R}$  una funció no lineal de la qual es vol trobar un zero simple  $\alpha \in J$ ; on  $J$  és un entorn de l'arrel; és a dir, es té l'equació:

$$f(x) = 0,$$

on  $\alpha$  és una solució.

Si la funció  $f$  és derivable, sense cap dubte el mètode més conegut i utilitzat és el mètode de Newton o també anomenat mètode de la tangent:

Donat  $x_0 \in J$ ,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n \geq 0.$$

Poden sorgir dificultats si ens trobem en un dels casos següents:

- $f$  no és derivable.
- $f$  no té una expressió analítica i, per tant, no es pot calcular la seva derivada.
- S'intenta resoldre un problema no lineal de valors frontera:

$$y'' = h(x, y, y'), \quad a \leq x \leq b,$$

amb condicions frontera  $y(a) = A$ ,  $y(b) = B$  i on  $h$  és una funció no lineal. El mètode del tir no lineal transforma el problema enunciat en un problema de valor inicial amb un paràmetre  $t = y'(a)$  i la resolució d'una equació en  $x = b$ , on no es coneix la derivada.

Quan no es pot fer servir la derivada, es fa servir l'operador de primer ordre diferències dividides definit per

$$[x, y]_f = \frac{f(y) - f(x)}{y - x}, \quad (2.1.1)$$

on  $y \neq x$  i  $x, y \in J$ .

Si la funció  $f$  és derivable en  $J$ , dues observacions:

- $[x, y]_f = f'(c)$  on  $c$  està entre  $x$  i  $y$ .
- $[x, x]_f = f'(x)$ .

## 2.2 Mètodes iteratius que utilitzen l'operador de diferències dividides

Els mètodes més coneguts per resoldre equacions no lineals que fan servir diferències dividides són:

- El mètode de la Secant [4, 9, 16]: Donats  $x_{n-1}$  i  $x_n \in J$ ,

$$x_{n+1} = \phi_1(x_n; x_{n-1}) = x_n - [x_{n-1}, x_n]_f^{-1} f(x_n), \quad n \geq 0. \quad (2.2.1)$$

- EL mètode de Kurxatov [21, 29]: Donat  $x_{n-1}, x_n \in J$ ,

$$x_{n+1} = \phi_2(x_n; x_{n-1}) = x_n - [x_{n-1}, 2x_n - x_{n-1}]_f^{-1} f(x_n), \quad n \geq 0. \quad (2.2.2)$$

- EL mètode de Steffensen [3, 2, 4, 30]: Donat  $x_n \in J$ ,

$$x_{n+1} = \phi_3(x_n) = x_n - [x_n, x_n + f(x_n)]_f^{-1} f(x_n), \quad n \geq 0. \quad (2.2.3)$$

Els dos primers algorismes, (1.4) i (1.5), el mètode de la Secant i Kurxatov, són mètodes iteratius amb memòria mentre que el d'Steffensen (1.6) no té memòria.

La notació  $\phi(x_n; x_{n-1})$  indica que  $x_{n-1}$  i  $x_n$  s'utilitzen en el càlcul de  $x_{n+1}$  (amb memòria) mentre que  $\phi(x_n)$  indica que  $x_n$  és l'únic que té contribució efectiva en aquests càlcul (sense memòria).

## 2.3 Definicions de l'ordre de convergència

A partir de les aproximacions inicials  $x_{-1}, x_0 \in J$  per  $\phi_1$  i  $\phi_2$  i  $x_0 \in J$  per  $\phi_3$ , els mètodes iteratius anteriors donen lloc a una successió infinita:

$$\mathcal{S} = \{x_1, x_2, \dots, x_n, x_{n+1}, \dots\}.$$

La successió  $\mathcal{S}$  serà interessant en tant que sigui convergent cap a la solució del problema. És a dir, se suposa que  $\mathcal{S}$  té límit  $\alpha$  i es pot escriure de la forma següent

$$\lim_{n \rightarrow \infty} x_n = \alpha.$$



Tenint en consideració que, en realitat, només caldrà un subconjunt finit de  $\mathcal{S}$ , és a dir  $S = \{x_1, x_2, \dots, x_I\}$ , si es defineix  $e_n = x_n - \alpha$  on  $n = 1, \dots, I$ , s'obté l'error absolut,  $|e_n|$ , en cada pas, i el mètode iteratiu s'aturarà quan s'acompleixi el criteri d'aturada següent:

$$|e_n| < \varepsilon,$$

on  $\varepsilon$  és prou petit. Dit d'una manera més precisa:

$$|x_I - \alpha| > \varepsilon, \quad |x_{I+1} - \alpha| < \varepsilon.$$

S'observa que  $|e_{I+1}| = |x_{I+1} - \alpha|$  és el primer error que compleix el criteri de tall.

**Definició 2.3.1.** Traub (1964, [33]). Donada la successió  $\mathcal{S}$  convergent cap a  $\alpha$ , si  $\exists \rho \in \mathbb{R}^+$ ,  $\rho \geq 1$ , i  $\exists \mathcal{C} > 0$  tal que

$$\lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|^\rho} = \mathcal{C}, \quad (2.3.1)$$

llavors,  $\rho$  és l'ordre de convergència de  $\mathcal{S}$  i  $\mathcal{C}$  s'anomena constant asimptòtica de l'error.

Quan es parla de l'ordre local de convergència d'un mètode iteratiu al voltant d'una arrel  $\alpha$ , s'està referint a l'ordre de la successió  $\mathcal{S}$  generada per l'esquema iteratiu  $\phi$ . Es pot explicitar fent servir la notació següent:

$$\lim_{n \rightarrow \infty} \frac{|\phi(x_n) - \alpha|}{|x_n - \alpha|^\rho} = \mathcal{C}.$$

A causa que l'ordre de  $\phi$  depèn de  $f$ , aquest ordre pot ser diferent per a diferents classes de  $f$ . Igualment, l'ordre d'un mètode pot canviar si s'aplica el mètode al càlcul de diferents arrels de la mateixa funció  $f$ .

Per evitar termes d'ordre superior en algunes relacions, que no influeixen en l'ordre de convergència, emprem la notació de Landau ( $o$  i  $\mathcal{O}$ ). Si  $a_n$  i  $b_n$  són successions i  $b_n/a_n \rightarrow K$  quan  $n \rightarrow \infty$ , on  $K \neq 0$ , s'escriu

$$b_n = \mathcal{O}(a_n) \text{ o } a_n \sim b_n.$$

Si  $b_n/a_n \rightarrow 0$  quan  $n \rightarrow \infty$  s'escriu  $b_n = o(a_n)$ ; dit de diferent manera,  $a_n$  és dominat per  $b_n$  asimptòticament. Aquest enfocament simplifica l'anàlisi de la convergència.

Un concepte que es fa servir sovint en aquest treball és l'expressió de l'error mitjançant una equació en diferències. De (2.3.1), per a  $n$  suficientment gran, es té,

$$|e_{n+1}| = \mathcal{C}|e_n|^\rho + \mathcal{O}(|e_n|^{\rho+1}). \quad (2.3.2)$$

L'equació (2.3.2) és estàndard per a mètodes iteratius sense memòria com per exemple  $\phi_3$ , equació (2.2.3), amb  $\rho \in \mathbb{Z}^+$ ,  $\rho \geq 1$ .

Per a mètodes iteratius amb memòria del tipus  $x_{n+1} = \Phi(x_n; x_{n-1})$ , l'equació en diferències de l'error vé donada per

$$|e_{n+1}| = \mathcal{C}|e_n|^a |e_{n-1}|^b + o(|e_n|^a |e_{n-1}|^b), \quad \text{on } a, b \in \mathbb{Z}^+ \quad (2.3.3)$$

En aquest cas, l'error és l'única arrel positiva de l'equació indicial [32]  $t^2 - at - b = 0$ .

**Definició 2.3.2.** Wall (1956, [34]). Una darrera definició de l'ordre vé proposada per Wall

$$\rho = \lim_{n \rightarrow \infty} \frac{\log |e_{n+1}|}{\log |e_n|} \quad (2.3.4)$$

Es fa servir més endavant per comprovar d'una manera computacional l'ordre dels mètodes iteratius presentats.

**Nota 1.** Una manera d'expressar el número aproximat de decimals correctes que representa  $x_n$  respecte  $\alpha$  és

$$d_n = -\log_{10} |e_n| \quad (2.3.5)$$

De la igualtat (2.3.5) de la nota 1, la definició (2.3.4) també s'expresa com

$$\rho = \lim_{n \rightarrow \infty} \frac{d_{n+1}}{d_n}. \quad (2.3.6)$$

De (2.3.6) es dedueix que per a  $n$  suficientment gran, o  $x_n$  suficientment pròxim a  $\alpha$ , es té

$$d_{n+1} \approx \rho d_n, \quad (2.3.7)$$

de manera que el número de decimals correctes en cada pas del procés iteratiu és, aproximadament,  $\rho$  cops el número de decimals correctes calculats en el pas anterior, on  $\rho$  és l'ordre local de convergència.

## 2.4 Eficiència computacional

En aquest treball es presenten diversos mètodes iteratius que es construeixen a partir de  $\phi_i$ ,  $i = 1 \div 3$ , considerant variants, pseudo-composicions i congelacions de l'operador.

A l'hora de trobar quin és el millor algorisme iteratiu per resoldre una equació s'haurà de considerar el concepte d'eficiència.

**Definició 2.4.1.** Es defineix l'índex d'eficiència computacional  $CEI$  [15] per

$$CEI = \rho^{1/\epsilon},$$

on  $\rho$  és l'ordre local de convergència i  $\mathcal{C}$  és el cost computacional, expressat en productes, en considerar l'evaluació de les funcions que entren en una iteració del mètode i les operacions que s'han d'efectuar en aquesta iteració.

El cost computacional ve donat per l'equació:

$$\mathcal{C} = a\mu + p + q\ell, \quad (2.4.1)$$

on  $a$  és el nombre d'evaluacions de la funció  $f$ ,  $q$  és el nombre de quocients i  $p$  el nombre de productes, per iteració. A més a més,  $\mu$  és el cost d'una avaluació de la funció expressat en productes i  $\ell$  és l'equivalent en productes d'un quocient.

**Nota 2.** Es poden comparar mètodes iteratius mitjançant la comparació de la seva eficiència. Per tal objectiu farem servir els  $CEI$ 's i utilitzant el quocient:

$$R_{i,j} = \frac{\log CEI_i}{\log CEI_j} = \frac{\log \rho_i}{\log \rho_j} \frac{\mathcal{C}_j}{\mathcal{C}_i} \quad (2.4.2)$$

**Nota 3.** No es consideren mètodes iteratius composts amb ells mateixos ja que l'eficiència és la mateixa: si es pren  $\phi_i$  amb ordre  $\rho$  i cost  $\mathcal{C}$ , quan es considera  $\phi_i^2 = \phi_i \circ \phi_i$  es té ordre  $\rho^2$  i cost computacional  $2\mathcal{C}$ ; per tant,

$$CEI(\phi_i) = \rho^{1/\mathcal{C}} = (\rho^2)^{1/2\mathcal{C}} = CEI(\phi_i^2).$$

**Nota 4.** Si s'estudia la pseudo-composició de dos mètodes, per exemple  $\phi_a(x_n; x_{n-1})$  i  $\phi_b(x_n; x_{n-1})$ , es té  $\phi_b \diamond \phi_a$  definit per:

Donats  $x_{n-1}, x_n \in J$ ,

$$\begin{cases} y_n = \phi_a(x_n; x_{n-1}), \\ x_{n+1} = \phi_b(y_n; x_n). \end{cases}$$

Es té el doble de cost computacional però l'ordre no es suficient per obtenir mètodes eficients.

Veiem, per exemple, la pseudo-composició del mètode de la secant amb si mateix i també la pseudo-composició de Kurxatov, (2.2.1) i (2.2.2) respectivament:

$$\begin{aligned} \rho(\phi_1) = (1 + \sqrt{5})/2 &\rightarrow \rho(\phi_1 \diamond \phi_1) = 1 + \sqrt{2} \\ \rho(\phi_2) = 2 &\rightarrow \rho(\phi_2 \diamond \phi_2) = 1 + \sqrt{5} \end{aligned}$$

La demostració de l'ordre de convergència que presenten els mètodes iteratius estudiats es veurà més endavant.

Si es considera el temps expressat en productes, definim el factor temps,  $TF$ , com:

$$TF = \frac{1}{\ln CEI} = \frac{\mathcal{C}}{\ln \rho} \quad (2.4.3)$$

Altres autors ([25], [33]) utilitzen  $EI = \rho^{1/a}$ , cas particular de  $CEI = \rho^{1/\mathcal{C}} = \rho^{1/(a\mu + q\ell + p)}$ .

## 2.5 Ordres Computacionals de convergència

Es defineixen uns paràmetres o mesures de l'ordre per calcular computacionalment. En la primera secció es fa a partir de la definició d'ordre de Traub (2.3.1) per tal de definir COC, ACOC, ECOOC i PCOC, mentre que posteriorment es fa servir la definició de Wall (2.3.4) per definir CLOC, ACLOC, ECLOC i PCLOC.

### 2.5.1 COC

L'ordre de convergència computacional (COC, *Computational Order of Convergence*) d'una successió  $\{x_n\}_{n \geq 1}$  es defineix com:

$$\bar{\rho}_n = \frac{\ln |e_n/e_{n-1}|}{\ln |e_{n-1}/e_{n-2}|}, \quad n \geq 3, \quad (2.5.1)$$

on  $e_n = x_n - \alpha$  i  $x_{n-2}$ ,  $x_{n-1}$  i  $x_n$  són tres iteracions consecutives. [35]

L'ordre computacional de convergència s'utilitza per verificar numèricament l'ordre de convergència dels mètodes iteratius presentats, on l'ordre de convergència ha estat estudiat teòricament.

Un dels majors inconvenients del COC és que es necessita saber l'arrel exacta  $\alpha$ , que habitualment no se sap.

### 2.5.2 ACOC

L'ordre de convergència computacional aproximat (ACOC, *Approximated Computational Order of Convergence*) d'una successió  $\{x_n\}_{n \geq 1}$  es defineix com:

$$\hat{\rho}_n = \frac{\ln |\hat{e}_n/\hat{e}_{n-1}|}{\ln |\hat{e}_{n-1}/\hat{e}_{n-2}|}, \quad n \geq 4, \quad (2.5.2)$$

on  $\hat{e}_n = x_n - x_{n-1}$ .

La relació entre  $\hat{e}_n$  i  $e_n$  es troba a [22] i és la següent:

$$e_n \sim \left( \frac{\hat{e}_n}{\hat{e}_{n-1}} \right)^{\rho^2/(\rho-1)}, \quad (2.5.3)$$

on la notació  $a_n \sim b_n$  equival a  $\lim_{n \rightarrow +\infty} \frac{b_n}{a_n} = K \neq 0$ .

### 2.5.3 ECOOC

Per tal de calcular l'error sense tenir en compte el valor d' $\alpha$ . Podem utilitzar tres iterats consecutius,  $x_n$ ,  $x_{n-1}$  i  $x_{n-2}$ , per tal d'aproximar  $\alpha$ .

Utilitzant l'extrapolació d'Aitken [1] l'aproximació d' $\alpha$  es pot aproximar de la següent forma:

$$\tilde{\alpha}_n = x_n - \frac{(\delta x_{n-1})^2}{\delta^2 x_{n-2}}, \quad n \geq 2, \quad (2.5.4)$$

on  $\delta x_k = x_{k+1} - x_k$  i  $\delta^2 x_{k-1} = \delta(\delta x_{k-1}) = x_{k+1} - 2x_k + x_{k-1}$ .

A partir de (2.5.4) podem definir una aproximació de l'error  $\tilde{e}_n = x_n - \tilde{\alpha}_n$  i el nou ordre de convergència (ECOC, *Extrapolated Computational Order of Convergence*), per tant, es defineix com:

$$\tilde{\rho}_n = \frac{\ln |\tilde{e}_n / \tilde{e}_{n-1}|}{\ln |\tilde{e}_{n-1} / \tilde{e}_{n-2}|}, \quad n \geq 5. \quad (2.5.5)$$

En [22] també es pot trobar la relació entre  $\tilde{e}_n$  i  $e_n$ , i és:

$$e_n \sim (\tilde{e}_n)^{\rho^2 / (\rho - 1)}. \quad (2.5.6)$$

## 2.5.4 PCOC

A partir del teorema del valor mig,  $f(x_n) - f(\alpha) = f'(c)(x_n - \alpha)$ , on  $c$  està entre  $x_n$  i  $\alpha$ . De la mateixa manera

$$f(x_{n-1}) = f'(d)(x_{n-1} - \alpha),$$

on  $d$  està entre  $x_{n-1}$  i  $\alpha$ . L'ordre de convergència computacional de Petković (2011,[26]) (PCOC, *Petković Computational Order of Convergence*) es pot abreujar amb la forma

$$\check{\rho}_n = \frac{\log |\check{e}_n|}{\log |\check{e}_{n-1}|}, \text{ on } \check{e}_n = \frac{f(x_n)}{f(x_{n-1})} \text{ i } n \geq 3. \quad (2.5.7)$$

La substitució de  $\frac{e_n}{e_{n-1}}$  en (2.5.1) és a causa de les igualtats següents:

$$\check{e}_n = \frac{f(x_n)}{f(x_{n-1})} = \frac{f'(c) e_n}{f'(d) e_{n-1}} \approx \frac{e_n}{e_{n-1}},$$

a causa que per  $n$  suficientment gran, es té  $\frac{f'(c)}{f'(d)} \approx 1$ , expressió aproximada de (2.5.1)

A més a més, de [18], es té

$$e_n \sim (\check{e}_n)^{\rho / (\rho - 1)}. \quad (2.5.8)$$

Fent servir la definició d'ordre de Wall (2.3.4), els càlculs es simplifiquen com es veu a continuació.

## 2.5.5 Definicions

Aquests paràmetres s'han introduït per Grau-Sánchez et al. (2012, [18]).

- L'ordre local de convergència computacional (CLOC, *Computational Local Order of Convergence*) es defineix com:

$$\bar{\lambda}_n = \frac{\log |e_n|}{\log |e_{n-1}|}, \quad n \geq 2. \quad (2.5.9)$$

- En el cas de ACLOC  $\hat{\lambda}_n$  (*Approximated Computational Local Order of Convergence*), l'error  $e_n$  se substitueix pel paràmetre  $\hat{e}_n = x_n - x_{n-1}$ . És a dir,

$$\hat{\lambda}_n = \frac{\log |\hat{e}_n|}{\log |\hat{e}_{n-1}|}, \quad n \geq 3. \quad (2.5.10)$$

- En el cas de ECLOC  $\tilde{\lambda}_n$  (*Extrapolated Computational Local Order of Convergence*) es pren  $\tilde{e}_n = x_n - \tilde{\alpha}_n$  en lloc de  $e_n$ , on  $\tilde{\alpha}_n$  està definit a (2.5.4). En definitiva,

$$\tilde{\lambda}_n = \frac{\log |\tilde{e}_n|}{\log |\tilde{e}_{n-1}|}, \quad n \geq 4. \quad (2.5.11)$$

- En el cas de PCLOC  $\check{\lambda}_n$  (*Petkovic Computational Local Order of Convergence*) sense el coneixement d' $\alpha$  a partir de  $f(x_n)$  i  $f(x_{n-1})$  es defineix per:

$$\check{\lambda}_n = \frac{\log |f(x_n)|}{\log |f(x_{n-1})|}, \quad n \geq 2. \quad (2.5.12)$$

## 2.6 Criteris de tall

A partir de la definició  $d_n = -\log_{10} |e_n|$  presentada a (2.3.5) que determina el número aproximat de decimals correctes en la iteració  $n$ -èssima i les relacions entre  $e_n$  respecte  $\hat{e}_n$ ,  $\tilde{e}_n$  i  $\check{e}_n$  descrites a (2.5.2), (2.5.3) i (2.5.4), es defineix el número de decimals correctes per a cada mesura de l'ordre de convergència de la següent manera:

$$\hat{d}_n = -\frac{\rho^2}{\rho - 1} \log_{10} \left| \frac{\hat{e}_n}{\hat{e}_{n-1}} \right|, \quad (2.6.1)$$

$$\tilde{d}_n = -\frac{\rho^2}{2\rho - 1} \log_{10} |\tilde{e}_n|, \quad (2.6.2)$$

$$\check{d}_n = -\frac{\rho}{\rho - 1} \log_{10} |\check{e}_n|. \quad (2.6.3)$$

A partir de la relació  $d_{n+1} \approx \rho d_n$ , presentada a (2.3.7) i les definicions (2.3.5), (2.6.1), (2.6.2) i (2.6.3) es presenten els diferents criteris de tall, on  $\eta$  és la precisió prefixada. Es recorda que  $S = \{x_1, \dots, x_I\} \subset \mathcal{S}$ :

$$d_I < \eta \quad \text{i} \quad \rho d_I > \eta, \quad (2.6.4)$$

o en els altres com:

$$\hat{d}_I < \eta \quad \text{i} \quad \frac{\rho^2}{\rho - 1} \hat{d}_I > \eta, \quad (2.6.5)$$

$$\tilde{d}_I < \eta \quad \text{i} \quad \frac{\rho^2}{2\rho - 1} \tilde{d}_I > \eta, \quad (2.6.6)$$

$$\check{d}_I < \eta \quad \text{i} \quad \frac{\rho}{\rho - 1} \check{d}_I > \eta. \quad (2.6.7)$$

## 2.7 Aritmètiques

Donat un mètode iteratiu MI, es pot fer servir una aritmètica fixa per fer el càlcul de tots els iterats de la successió  $\mathcal{S}$  o fer servir una aritmètica adaptativa que canvia en el càlcul de cada iterat de la successió  $\mathcal{S}$ .

Si se suposa que  $\alpha$  és coneguda i es considera un mètode iteratiu  $\Phi$  tal que  $x_i = \Phi(x_{i-1}; x_{i-2})$  i amb error  $e_i = x_i - \alpha$ , una predicció del nombre de decimals correctes en el càlcul de  $x_{i+1}$  és

$$m_{i+1} = [\rho (-\log_{10} |e_i| + k)], \quad (2.7.1)$$

on  $[\cdot]$  és la funció part entera i  $k$  un valor heurístic que ajusta el nombre de decimals per tal de donar una predicció lleugerament superior i assegurar que el nou càlcul serà representable en l'aritmètica predita.

Si  $\alpha$  és desconeguda, fent servir la relació entre els diferents paràmetres d'error descrits a (2.5.2), (2.5.3) i (2.5.4), es defineixen:

$$\hat{m}_{i+1} = \left[ \frac{\rho^3}{\rho - 1} \left( -\log_{10} \left| \frac{\hat{e}_i}{\hat{e}_{i-1}} \right| + k \right) \right], \quad (2.7.2)$$

$$\tilde{m}_{i+1} = \left[ \frac{\rho^3}{2\rho - 1} (-\log_{10} |\tilde{e}_i| + k) \right], \quad (2.7.3)$$

$$\check{m}_{i+1} = \left[ \frac{\rho^2}{\rho - 1} (-\log_{10} |\check{e}_i| + k) \right]. \quad (2.7.4)$$

Es pren  $m_{i+1}$  com la longitud de la mantissa per al nou càlcul de la iteració en el cas d' $\alpha$  desconeguda. En cas contrari, la longitud de la mantissa es pren com  $\hat{m}_{i+1}$ ,  $\tilde{m}_{i+1}$  o  $\check{m}_{i+1}$ .





## Capítol 3

# Mètodes Iteratius

### 3.1 Mètodes iteratius

Aquí es presenta un petit recordatori de la nomenclatura dels mètodes iteratius ja introduïts.

Mètode iteratiu	Abreviació	Nomenclatura	Equació
Mètode de la Secant	SM	$\phi_1$	(2.2.1)
Mètode de Kurxatov	KM	$\phi_2$	(2.2.2)
Mètode de Steffensen	StM	$\phi_3$	(2.2.3)

#### 3.1.1 Mètodes utilitzant pseudo-composició

Els mètodes iteratius de dos passos que s'estudien mitjançant la pseudo-composició dels mètodes de SM, KM i StM són:

$$\phi_4 = \phi_2 \diamond \phi_1 \quad SKM \begin{cases} y_n = \phi_1(x_n; x_{n-1}) &= x_n - [x_{n-1}, x_n]_f^{-1} f(x_n), \\ x_{n+1} = \phi_2(y_n; x_n) &= y_n - [x_n, 2y_n - x_n]_f^{-1} f(y_n). \end{cases} \quad (3.1.1)$$

$$\phi_5 = \phi_3 \diamond \phi_1 \quad SStM \begin{cases} y_n = \phi_1(x_n; x_{n-1}) &= x_n - [x_{n-1}, x_n]_f^{-1} f(x_n), \\ x_{n+1} = \phi_3(y_n) &= y_n - [y_n, y_n + f(y_n)]_f^{-1} f(y_n). \end{cases} \quad (3.1.2)$$

$$\phi_6 = \phi_1 \diamond \phi_2 \quad KSM \begin{cases} y_n = \phi_2(x_n; x_{n-1}) &= x_n - [x_{n-1}, 2x_n - x_{n-1}]_f^{-1} f(x_n), \\ x_{n+1} = \phi_1(y_n; x_n) &= y_n - [x_n, y_n]_f^{-1} f(y_n). \end{cases} \quad (3.1.3)$$

$$\phi_7 = \phi_3 \diamond \phi_2 \quad KStM \begin{cases} y_n = \phi_2(x_n; x_{n-1}) & = x_n - [x_{n-1}, 2x_n - x_{n-1}]_f^{-1} f(x_n), \\ x_{n+1} = \phi_3(y_n) & = y_n - [y_n, y_n + f(y_n)]_f^{-1} f(y_n). \end{cases} \quad (3.1.4)$$

$$\phi_8 = \phi_1 \diamond \phi_3 \quad StSM \begin{cases} y_n = \phi_3(x_n) & = x_n - [x_n, x_n + f(x_n)]_f^{-1} f(x_n), \\ x_{n+1} = \phi_1(y_n; x_n) & = y_n - [x_n, y_n]_f^{-1} f(y_n). \end{cases} \quad (3.1.5)$$

$$\phi_9 = \phi_2 \diamond \phi_3 \quad StKM \begin{cases} y_n = \phi_3(x_n) & = x_n - [x_n, x_n + f(x_n)]_f^{-1} f(x_n), \\ x_{n+1} = \phi_2(y_n; x_n) & = y_n - [x_n, 2y_n - x_n]_f^{-1} f(y_n). \end{cases} \quad (3.1.6)$$

### 3.1.2 Mètodes amb operador congelat

**Definició 3.1.1. Congelat** [14] Es diu que un mètode de dos passos és congelat quan l'operador es repeteix; és a dir, donats  $x_{n-1}, x_n \in J$  o  $x_n \in J$  i construït l'operador  $\Theta$ , on  $n \geq 0$ , es considera:

$$\begin{cases} y_n = x_n - \Theta^{-1} f(x_n), \\ x_{n+1} = y_n - \Theta^{-1} f(y_n). \end{cases}$$

És un mètode congelat de dos passos a partir de l'esquema  $x_{n+1} = x_n - \Theta^{-1} f(x_n)$ .

Definim els operadors congelats dels mètodes de la SM, KM i StM com:

Mètode iteratiu	Operador Congelat
Congelat de la Secant	$\Theta_1 = [x_{n-1}, x_n]_f$
Congelat de Kurxatov	$\Theta_2 = [x_{n-1}, 2x_n - x_{n-1}]_f$
Congelat de Steffensen	$\Theta_3 = [x_n, x_n + f(x_n)]_f$

Definim ara els mètodes congelats dels mètodes SM, KM i StM seguint la nomenclatura FSM [27], FKM i FStM que es forma amb F de congelat (*frozen*) i l'abreviació del mètode utilitzada fins ara.

$$\phi_{10} \quad FSM \begin{cases} y_n = \phi_1(x_n; x_{n-1}) & = x_n - \Theta_1^{-1} f(x_n), \\ x_{n+1} & = y_n - \Theta_1^{-1} f(y_n). \end{cases} \quad (3.1.7)$$

$$\phi_{11} \quad FKM \begin{cases} y_n = \phi_2(x_n; x_{n-1}) & = x_n - \Theta_2^{-1} f(x_n), \\ x_{n+1} & = y_n - \Theta_2^{-1} f(y_n). \end{cases} \quad (3.1.8)$$

$$\phi_{12} \quad FStM \begin{cases} y_n = \phi_3(x_n) & = x_n - \Theta_3^{-1} f(x_n), \\ x_{n+1} & = y_n - \Theta_3^{-1} f(y_n). \end{cases} \quad (3.1.9)$$

### 3.1.3 Mètodes pseudo-composats amb si mateixos

Per tal de comprovar que no guanyem eficiència pseudo-composant els mètodes amb ells mateixos, presentem els següents mètodes i després analitzarem el seu ordre de convergència a la secció 3.3.

$$\phi_{13} = \phi_1 \diamond \phi_1 \quad SSM \quad \begin{cases} y_n = \phi_1(x_n; x_{n-1}) &= x_n - [x_{n-1}, x_n]_f^{-1} f(x_n), \\ x_{n+1} = \phi_1(y_n; x_n) &= y_n - [x_n, y_n]_f^{-1} f(y_n). \end{cases} \quad (3.1.10)$$

$$\phi_{14} = \phi_2 \diamond \phi_2 \quad KKM \quad \begin{cases} y_n = \phi_2(x_n; x_{n-1}) &= x_n - [x_n, 2x_{n-1} - x_n]_f^{-1} f(x_n), \\ x_{n+1} = \phi_2(y_n; x_n) &= y_n - [x_n, 2y_n - x_n]_f^{-1} f(y_n). \end{cases} \quad (3.1.11)$$

$$\phi_{15} = \phi_3 \circ \phi_3 \quad StStM \quad \begin{cases} y_n = \phi_3(x_n) &= x_n - [x_n, x_n + f(x_n)]_f^{-1} f(x_n), \\ x_{n+1} = \phi_3(y_n) &= y_n - [y_n, y_n + f(y_n)]_f^{-1} f(y_n). \end{cases} \quad (3.1.12)$$

### 3.1.4 Altres mètodes

Una altra manera de construir mètodes amb l'operador congelat i amb memòria es troben a Shakno [29] i Traub [31]. Donats  $y_{n-1}$  i  $x_n$ , el mètode de Shakno es defineix com:

$$\phi_{16} \quad ShM \quad \begin{cases} y_n &= y_{n-1} - [y_{n-1}, x_n]_f^{-1} f(y_{n-1}), \\ x_{n+1} &= y_n - [y_{n-1}, x_n]_f^{-1} f(y_n). \end{cases} \quad (3.1.13)$$

Donats  $y_{n-1}$  i  $x_n$ , el mètode de Traub es defineix mitjançant:

$$\phi_{17} \quad TM \quad \begin{cases} y_n &= x_n - [x_{n-1}, y_{n-1}]_f^{-1} f(x_n), \\ x_{n+1} &= x_n - [x_n, y_n]_f^{-1} f(x_n). \end{cases} \quad (3.1.14)$$

### 3.2 Estudi de l'ordre de convergència. Desenvolupaments

A partir de la fórmula de Taylor per a la funció  $f$  en un entorn de l'arrel  $\alpha$ , es té

$$\begin{aligned} f(z) &= f(\alpha) + f'(\alpha)(z - \alpha) + \frac{1}{2}f''(\alpha)(z - \alpha)^2 + \frac{1}{6}f'''(\alpha)(z - \alpha)^3 + \mathcal{O}(z - \alpha)^4 \\ &= f'(\alpha) (E + A_2E^2 + A_3E^3 + \mathcal{O}(E^4)) \end{aligned} \quad (3.2.1)$$

on  $E = z - \alpha$  i  $A_k = \frac{1}{k!}f'(\alpha)^{-1}f^{(k)}(\alpha)$ ,  $k \geq 2$ .

Definint  $e = x - \alpha$  i  $\varepsilon = y - \alpha$ , a partir de (3.2.1) es té

$$\begin{aligned} [x, y]_f &= \frac{f(y) - f(x)}{y - x} \\ &= f'(\alpha) \frac{1}{\varepsilon - e} (\varepsilon - e + A_2(\varepsilon^2 - e^2) + A_3(\varepsilon^3 - e^3) + \mathcal{O}(\varepsilon^4 - e^4)) \\ &= \Gamma \left( 1 + A_2(e + \varepsilon) + A_3(e^2 + e\varepsilon + \varepsilon^2) + \mathcal{O}_3(e, \varepsilon) \right), \end{aligned} \quad (3.2.2)$$

on  $\Gamma = f'(\alpha)$  i  $\mathcal{O}_3(e, \varepsilon) = \mathcal{O}(e^{p_1}\varepsilon^{p_2})$  amb  $p_i \in \mathbb{Z}$ ,  $0 \leq p_i \leq 3$  i  $p_1 + p_2 = 3$ .

Tenint en compte que per a  $|\beta| < 1$  es compleix

$$\frac{1}{1 + \beta} = 1 - \beta + \beta^2 + \mathcal{O}(\beta^3), \quad (3.2.3)$$

llavors, podem considerar  $[x, y]_f = \Gamma(1 + \beta)$ , on  $\beta = A_2(e + \varepsilon) + A_3(e^2 + e\varepsilon + \varepsilon^2) + \mathcal{O}_3(e, \varepsilon)$ ; si s'aplica (3.2.3) a (3.2.2), s'obté

$$[x, y]_f^{-1} = \Gamma^{-1} \left( 1 - A_2(\varepsilon + e) - A_3(\varepsilon^2 + e\varepsilon + e^2) + A_2^2(\varepsilon^2 + 2e\varepsilon + e^2) + \mathcal{O}_3(\varepsilon, e) \right) \quad (3.2.4)$$

Els desenvolupaments dels operadors que cal considerar per tal de calcular l'ordre local de convergència són:

$$\Theta_1^{-1} = [x_{n-1}, x_n]_f^{-1} = (1 - A_2(e_{n-1} + e_n) + o(e_{n-1}))\Gamma^{-1} \quad (3.2.5)$$

$$\Theta_2^{-1} = [x_{n-1}, 2x_n - x_{n-1}]_f^{-1} = (1 - 2A_2e_n - A_3e_{n-1}^2 + o(e_{n-1}^2))\Gamma^{-1} \quad (3.2.6)$$

$$\Theta_3^{-1} = [x_n, x_n + f(x_n)]_f^{-1} = (1 - A_2(2 + \Gamma)e_n + o(e_n))\Gamma^{-1} \quad (3.2.7)$$

Per tant, per a  $\phi_i$ ,  $i = 1, 2, 3$ , tenim les equacions d'error següents:

$$e_{n+1}^{(1)} = A_2e_{n-1}e_n + o(e_{n-1}e_n), \quad \text{amb } \rho_1 = (1 + \sqrt{5})/2, \quad (3.2.8)$$

$$e_{n+1}^{(2)} = A_2 e_n^2 + A_3 e_{n-1}^2 e_n + o(e_{n-1}^2 e_n, e_n^2), \quad \text{amb } \rho_2 = 2, \quad (3.2.9)$$

$$e_{n+1}^{(3)} = A_2(1 + \Gamma)e_n^2 + \mathcal{O}(e_n^3), \quad \text{amb } \rho_3 = 2. \quad (3.2.10)$$

A la secció 3.3 es veurà amb més detall aquests resultats.

### 3.3 Equacions dels errors i ordres locals de convergència

Restan  $\alpha$  als dos membres de cadascun dels mètodes iteratius presentats  $\phi_i$ ,  $1 \leq i \leq 3$  i tenint en consideració (3.2.1) i (3.2.5), (3.2.6) i (3.2.7) respectivament, es té:

**SM:** De  $\phi_1(x_n; x_{n-1}) = x_n - [x_{n-1}, x_n]_f^{-1} f(x_n)$ , s'en segueix

$$\begin{aligned} e_{n+1}^{(1)} &= e_n - (1 - A_2(e_{n-1} + e_n) + o(e_n))(e_n + A_2 e_n^2 + \mathcal{O}(e_n^3)) \\ &= A_2 e_{n-1} e_n + o(e_{n-1} e_n). \end{aligned} \quad (3.3.1)$$

L'equació de l'error associada a l'equació amb diferències (3.3.1) és  $t^2 - t - 1 = 0$  i l'arrel positiva d'aquesta equació és l'ordre; per tant, l'ordre local de convergència del mètode de la Secant és  $\rho_1 = (1 + \sqrt{5})/2$ .

**KM:** De  $\phi_2(x_n; x_{n-1}) = x_n - [x_{n-1}, 2x_n - x_{n-1}]_f^{-1} f(x_n)$  l'equació de l'error és

$$\begin{aligned} e_{n+1}^{(2)} &= e_n - (1 - 2A_2 e_n - A_3 e_{n-1}^2 + o(e_{n-1}^2))(e_n + A_2 e_n^2 + \mathcal{O}(e_n^3)) \\ &= A_2 e_n^2 + A_3 e_{n-1}^2 e_n + o(e_n^2, e_{n-1}^2 e_n), \end{aligned} \quad (3.3.2)$$

El segon sumand del segon membre de (3.3.2) té l'equació indicial  $t^2 - t - 2 = 0$ ; per tant, l'ordre local de convergència del mètode de Kurxatov és  $\rho_2 = 2$ , a causa dels dos termes:  $A_2 e_n^2$  i  $A_3 e_{n-1}^2 e_n$ .

**StM:** De  $\phi_3(x_n) = x_n - [x_n, x_n + f(x_n)]_f^{-1} f(x_n)$ , s'obté

$$\begin{aligned} e_{n+1}^{(3)} &= e_n - (1 - A_2(2e_n + \Gamma e_n) + o(e_n))(e_n + A_2 e_n^2 + \mathcal{O}(e_n^3)) \\ &= A_2(1 + \Gamma)e_n^2 + \mathcal{O}(e_n^3), \end{aligned} \quad (3.3.3)$$

i l'ordre local de convergència del mètode de Steffensen és  $\rho_3 = 2$ .

### 3.3.1 Pseudo-composats

Si es procedeix de la mateixa manera que en els tres primers algorismes bàsics es tenen les expressions dels errors següents:

**SKM:** De  $\phi_4 = \phi_2 \diamond \phi_1 = \begin{cases} y_n = \phi_1(x_n; x_{n-1}) &= x_n - [x_{n-1}, x_n]_f^{-1} f(x_n), \\ x_{n+1} = \phi_2(y_n; x_n) &= y_n - [x_n, 2y_n - x_n]_f^{-1} f(y_n), \end{cases}$  es té

$$\begin{aligned} \varepsilon_n^{(4)} &= \varepsilon_n^{(1)} = A_2 e_{n-1} e_n + o(e_{n-1} e_n), \\ e_{n+1}^{(4)} &= \varepsilon_n^{(1)} - (1 - 2A_2 \varepsilon_n^{(1)} - A_3 e_n^2 + o(e_n^2))(\varepsilon_n^{(1)} + o(\varepsilon_n^{(1)})) \\ &= 2A_2 \varepsilon_n^{(1)2} + o(\varepsilon_n^{(1)2}) \\ &= 2A_2^3 e_{n-1}^2 e_n^2 + o(e_{n-1}^2 e_n^2). \end{aligned} \quad (3.3.4)$$

L'expressió associada a (3.3.4) és  $t^2 - 2t - 2 = 0$ ; per tant, s'obté que l'ordre del mètode iteratiu  $\phi_4$  és  $\rho_4 = 1 + \sqrt{3}$ .

**SStM:** De  $\phi_5 = \phi_3 \diamond \phi_1 = \begin{cases} y_n = \phi_1(x_n; x_{n-1}) &= x_n - [x_{n-1}, x_n]_f^{-1} f(x_n), \\ x_{n+1} = \phi_3(y_n) &= y_n - [y_n, y_n + f(y_n)]_f^{-1} f(y_n), \end{cases}$  s'aconsegueix la següent expressió dels errors

$$\begin{aligned} \varepsilon_n^{(5)} &= \varepsilon_n^{(1)} = A_2 e_{n-1} e_n + o(e_{n-1} e_n), \\ e_{n+1}^{(5)} &= \varepsilon_n^{(1)} - (1 - A_2(2\varepsilon_n^{(1)} + \tilde{\varepsilon}_n^{(1)}) + \mathcal{O}(\varepsilon_n^{(1)}))(\varepsilon_n^{(1)} + A_2(\varepsilon_n^{(1)2})\mathcal{O}(\varepsilon_n^{(1)2})) \\ &= A_2(\varepsilon_n^{(1)} + \Gamma \varepsilon_n^{(1)})\varepsilon_n^{(1)} + \mathcal{O}(\varepsilon_n^{(1)2}) \\ &= (1 + \Gamma)A_2^3 e_{n-1}^2 e_n^2 + o(e_{n-1}^2 e_n^2). \end{aligned} \quad (3.3.5)$$

L'equació associada a (3.3.5) és  $t^2 - 2t - 2 = 0$ ; per tant, l'ordre de convergència del mètode  $\phi_5$  és  $\rho_5 = 1 + \sqrt{3}$ .

**KSM:** De  $\phi_6 = \phi_1 \diamond \phi_2 = \begin{cases} y_n = \phi_2(x_n; x_{n-1}) &= x_n - [x_{n-1}, 2x_n - x_{n-1}]_f^{-1} f(x_n), \\ x_{n+1} = \phi_1(y_n; x_n) &= y_n - [x_n, y_n]_f^{-1} f(y_n), \end{cases}$  s'obtenen les equacions en diferències següents:

$$\begin{aligned} \varepsilon_n^{(6)} &= \varepsilon_n^{(2)} = A_2 e_n^2 + A_3 e_{n-1}^2 e_n + o(e_n^2, e_{n-1} e_n), \\ e_{n+1}^{(6)} &= \varepsilon_n^{(2)} - (1 - A_2(e_n + \varepsilon_n^{(2)}) + o(e_n))(\varepsilon_n^{(2)} + o(\varepsilon_n^{(2)})) \\ &= A_2 e_n \varepsilon_n^{(2)} + o(e_n \varepsilon_n^{(2)}) \\ &= A_2^2 e_n^3 + A_2 A_3 e_{n-1}^2 e_n^2 + o(e_{n-1}^2 e_n^2). \end{aligned} \quad (3.3.6)$$

L'equació indicial associada a (3.3.6) és  $t^2 - 2t - 2 = 0$ ; per tant, l'esquema iteratiu  $\phi_6$  té ordre  $\rho_6 = 1 + \sqrt{3}$ .

**KStM:** De  $\phi_7 = \phi_3 \diamond \phi_2 = \begin{cases} y_n = \phi_2(x_n; x_{n-1}) & = x_n - [x_{n-1}, 2x_n - x_{n-1}]_f^{-1} f(x_n), \\ x_{n+1} = \phi_3(y_n) & = y_n - [y_n, y_n + f(y_n)]_f^{-1} f(y_n), \end{cases}$  es té

$$\begin{aligned} \varepsilon_n^{(7)} &= \varepsilon_n^{(2)} = A_2 e_n^2 + A_3 e_{n-1}^2 e_n + o(e_n^2, e_{n-1}^2 e_n), \\ e_{n+1}^{(7)} &= \varepsilon_n^{(2)} - (1 - A_2(2\varepsilon_n^{(2)} - \Gamma\varepsilon_n^{(2)}) + o(\varepsilon_n^{(2)}))(\varepsilon_n^{(2)} + A_2(\varepsilon_n^{(2)})^2 + o(\varepsilon_n^{(2)2})) \\ &= A_2(\varepsilon_n^{(2)2} + \varepsilon_n^{(2)}\Gamma\varepsilon_n^{(2)}) + o(\varepsilon_n^{(2)2}) \\ &= (A_2^3 e_n^4 + A_2 A_3^2 e_{n-1}^4 e_n^2 + 2A_2^2 A_3 e_{n-1}^2 e_n^3)(1 + \Gamma) + o(e_{n-1}^4 e_n^2). \end{aligned} \quad (3.3.7)$$

De (3.3.7) es té  $t^2 - 2t - 4 = 0$ ; per tant, l'ordre de l'algorisme  $\phi_7$  ve donat pel terme  $A_2 A_3^2 e_n^2 e_{n-1}^4$  i és  $\rho_7 = 1 + \sqrt{5}$ .

**StSM:** De  $\phi_8 = \phi_1 \diamond \phi_3 = \begin{cases} y_n = \phi_3(x_n) & = x_n - [x_n, x_n + f(x_n)]_f^{-1} f(x_n), \\ x_{n+1} = \phi_1(y_n; x_n) & = y_n - [x_n, y_n]_f^{-1} f(y_n), \end{cases}$  es poden expressar els errors següents:

$$\begin{aligned} \varepsilon_n^{(8)} &= \varepsilon_n^{(3)} = A_2(1 + \Gamma)e_n^2 + \mathcal{O}(e_n^3), \\ e_{n+1}^{(8)} &= A_2 e_n \varepsilon_n^{(3)} + \mathcal{O}(e_n \varepsilon_n^{(3)}) \\ &= A_2^2(1 + \Gamma)e_n^3 + \mathcal{O}(e_n^5). \end{aligned} \quad (3.3.8)$$

Per tant, per (3.3.8)  $\phi_8$  té ordre local de convergència  $\rho_8 = 3$ .

**StKM:** De  $\phi_9 = \phi_2 \diamond \phi_3 = \begin{cases} y_n = \phi_3(x_n) & = x_n - [x_n, x_n + f(x_n)]_f^{-1} f(x_n), \\ x_{n+1} = \phi_2(y_n; x_n) & = y_n - [x_n, 2y_n - x_n]_f^{-1} f(y_n), \end{cases}$  es té

$$\begin{aligned} \varepsilon_n^{(9)} &= \varepsilon_n^{(3)} = A_2(1 + \Gamma)e_n^2 + \mathcal{O}(e_n^3), \\ e_{n+1}^{(9)} &= A_2 \varepsilon_n^{(3)} + A_3 e_n^2 \varepsilon_n^{(3)} + \mathcal{O}(e_n^4) \\ &= A_2(1 + \Gamma)(A_2(1 + \Gamma) + A_3)e_n^4 + \mathcal{O}(e_n^4). \end{aligned} \quad (3.3.9)$$

Llavors, l'ordre local de  $\phi_9$  trobat a (3.3.9) és  $\rho_9 = 4$ .

### 3.3.2 Congelats

L'operador de diferències dividides és el mateix en els dos passos.

**FSM:** Congelant el mètode de la secant un cop:  $\phi_{10} = \begin{cases} y_n = \phi_1(x_n; x_{n-1}) & = x_n - \Theta_1^{-1} f(x_n), \\ x_{n+1} & = y_n - \Theta_1^{-1} f(y_n), \end{cases}$

on  $\Theta_1 = [x_{n-1}, x_n]$ , es dedueix:

$$\begin{aligned}
\varepsilon_n^{(10)} &= \varepsilon_n^{(1)} = A_2 e_{n-1} e_n + o(e_{n-1} e_n), \\
e_{n+1}^{(10)} &= \varepsilon_n^{(1)} - (1 - A_2(e_{n-1} + e_n) + o(e_{n-1}))(\varepsilon_n^{(1)} + o(\varepsilon_n^{(1)})) \\
&= A_2 e_{n-1} \varepsilon_n^{(1)} + o(e_{n-1} \varepsilon_n^{(1)}) \\
&= A_2 e_{n-1}^2 e_n + o(e_{n-1}^2 e_n).
\end{aligned} \tag{3.3.10}$$

L'equació indicial de (3.3.10) és  $t^2 - t - 2 = 0$ ; per tant, l'ordre local de convergència de  $\phi_{10}$  és  $\rho_{10} = 2$ . És a dir, en aquest cas l'ordre local és almenys quadràtic.

**FKM:** Congelant el mètode de Kurxatov:  $\phi_{11} = \begin{cases} y_n = \phi_2(x_n; x_{n-1}) &= x_n - \Theta_2^{-1} f(x_n), \\ x_{n+1} &= y_n - \Theta_2^{-1} f(y_n), \end{cases}$  on  $\Theta_2 = [x_{n-1}, 2x_n - x_{n-1}]$  es poden expressar els errors següents:

$$\begin{aligned}
\varepsilon_n^{(11)} &= \varepsilon_n^{(2)} = A_2 e_n^2 + A_3 e_{n-1}^2 e_n + o(e_n^2, e_{n-1}^2 e_n), \\
e_{n+1}^{(11)} &= \varepsilon_n^{(2)} - (1 - 2A_2 e_n - A_3 e_{n-1}^2 + o(e_{n-1}^2))(\varepsilon_n^{(2)} + o(\varepsilon_n^{(2)})) \\
&= 2A_2 e_n(\varepsilon_n^{(2)} + A_3 e_{n-1}^2 \varepsilon_n^{(2)} + o(e_{n-1}^2 \varepsilon_n^{(2)})) \\
&= A_3^2 e_{n-1}^4 e_n + o(e_{n-1}^4 e_n)
\end{aligned} \tag{3.3.11}$$

L'equació associada a (3.3.11) és  $t^2 - t - 4 = 0$ ; per tant, l'ordre de convergència de  $\phi_{11}$  és  $\rho_{11} = (1 + \sqrt{17})/2$ .

**FStM:** Congelant el mètode de Steffensen:  $\phi_{12} = \begin{cases} y_n = \phi_3(x_n) &= x_n - \Theta_3^{-1} f(x_n), \\ x_{n+1} &= y_n - \Theta_3^{-1} f(y_n), \end{cases}$  on  $\Theta_3 = [x_n, x_n + f(x_n)]$ . Es té:

$$\begin{aligned}
\varepsilon_n^{(12)} &= \varepsilon_n^{(3)} = A_2(1 + \Gamma)e_n^2 + \mathcal{O}(e_n^3), \\
e_{n+1}^{(12)} &= \varepsilon_n^{(3)}(1 - A_2(2 + \Gamma)e_n + o(e_n))(\varepsilon_n^{(3)} + o(\varepsilon_n^{(3)})) \\
&= A_2(2 + \Gamma)e_n + \varepsilon_n^{(3)} + o(e_n \varepsilon_n^{(3)}) \\
&= A_2^2(1 + \Gamma)(2 + \Gamma)e_n^3 + o(e_n^3)
\end{aligned} \tag{3.3.12}$$

De (3.3.12), l'ordre local de convergència de  $\phi_{12}$  és  $\rho_{12} = 3$ .



### 3.3.3 Pseudo-composats amb si mateixos

**SSM:** De  $\phi_{13} = \phi_1 \diamond \phi_1 = \begin{cases} y_n = \phi_1(x_n; x_{n-1}) &= x_n - [x_{n-1}, x_n]_f^{-1} f(x_n), \\ x_{n+1} = \phi_1(y_n; x_n) &= y_n - [x_n, y_n]_f^{-1} f(y_n), \end{cases}$  se segueix:

$$\begin{aligned} \varepsilon_n^{(13)} &= \varepsilon_n^{(1)} = A_2 e_{n-1} e_n + o(e_{n-1} e_n), \\ e_{n+1}^{(13)} &= A_2 e_n \varepsilon_n^{(1)} + o(e_n \varepsilon_n^{(1)}) \\ &= A_2^2 e_{n-1} e_n^2 + o(e_{n-1} e_n^2). \end{aligned} \quad (3.3.13)$$

L'equació associada a (3.3.13) és  $t^2 - 2t - 1 = 0$ ; per tant, l'ordre del mètode  $\phi_{13}$  és  $\rho_{13} = 1 + \sqrt{2}$ .

**KKM:** De  $\phi_{14} = \phi_2 \diamond \phi_2 = \begin{cases} y_n = \phi_2(x_n; x_{n-1}) &= x_n - [x_n, 2x_{n-1} - x_n]_f^{-1} f(x_n), \\ x_{n+1} = \phi_2(y_n; x_n) &= y_n - [x_n, 2y_n - x_n]_f^{-1} f(y_n), \end{cases}$  es poden expressar els errors següents:

$$\begin{aligned} \varepsilon_n^{(14)} &= \varepsilon_n^{(2)} = A_2 e_n^2 + A_3 e_{n-1}^2 e_n + \mathcal{O}(e_n^2, e_{n-1}^2 e_n), \\ e_{n+1}^{(14)} &= \varepsilon_n^{(2)} - (1 - 2A_2 \varepsilon_n^{(2)}) + o(\varepsilon_n^{(2)}) (\varepsilon_n^{(2)} + A_2 \varepsilon_n^{(2)2} + o(\varepsilon_n^{(2)2})) \\ &= A_2 \varepsilon_n^{(2)2} + o(\varepsilon_n^{(2)2}) \\ &= A_2 (A_2^2 e_n^4 + A_3^2 e_{n-1}^2 e_n^2 + 2A_2 A_3 e_n^3 e_{n-1}) + o(\varepsilon_n^{(2)2}) \\ &= A_2 A_3^2 e_{n-1}^2 e_n^2 + o(e_{n-1}^2 e_n^2). \end{aligned} \quad (3.3.14)$$

L'equació indicial per (3.3.14) és  $t^2 - 2t - 4 = 0$ ; per tant  $\rho_{14} = 1 + \sqrt{5}$ .

**StStM:** De  $\phi_{15} = \phi_3 \circ \phi_3 = \begin{cases} y_n = \phi_3(x_n) &= x_n - [x_n, x_n + f(x_n)]_f^{-1} f(x_n), \\ x_{n+1} = \phi_3(y_n) &= y_n - [y_n, y_n + f(y_n)]_f^{-1} f(y_n). \end{cases}$  es té

$$\begin{aligned} \varepsilon_n^{(15)} &= \varepsilon_n^{(3)} = A_2(1 + \Gamma) e_n^2 + \mathcal{O}(e_n^3), \\ e_{n+1}^{(15)} &= A_2(1 + \Gamma) \varepsilon_n^{(3)2} + \mathcal{O}(\varepsilon_n^{(3)2}) \\ &= A_2^2(1 + \Gamma)^2 e_n^4 + \mathcal{O}(e_n^4). \end{aligned} \quad (3.3.15)$$

L'esquema iteratiu  $\phi_{15}$  dóna lloc a la composició de  $\phi_3$  amb ell mateix ja que no té memòria i és exactament el quadrat de StM (3.3.3). És a dir, per a (3.3.15)  $\rho_{15} = 4$ .

### 3.3.4 Altres mètodes

A continuació els mètodes de dos passos  $\phi_{16}$  i  $\phi_{17}$  [33, 29] amb un ordre de convergència local  $1 + \sqrt{2}$ .

**ShM:** De

$$\phi_{16} = \begin{cases} y_n &= y_{n-1} - [y_{n-1}, x_n]_f^{-1} f(y_{n-1}), \\ x_{n+1} &= y_n - [y_{n-1}, x_n]_f^{-1} f(y_n), \end{cases}$$

on el primer pas és el mètode de la Secant (2.2.1) amb  $y_{n-1}$  com a punt i  $[x_{n-1}, x_n]$  com a operador. Es dedueix:

$$\begin{aligned}
\varepsilon_n^{(16)} &= \varepsilon_{n-1} - (1 - A_2(\varepsilon_{n-1} + e_n) + o(\varepsilon_{n-1}))(\varepsilon_{n-1} + A_2\varepsilon_{n-1}^2 + o(\varepsilon_{n-1}^3)) \\
&= \varepsilon_{n-1} - \varepsilon_{n-1} - A_2\varepsilon_{n-1}^2 + A_2\varepsilon_{n-1}^2 + A_2e_n\varepsilon_{n-1} + o(\varepsilon_{n-1}^2) \\
&= A_2e_n\varepsilon_{n-1} + o(e_n\varepsilon_{n-1}), \\
e_{n+1}^{(16)} &= \varepsilon_n - (1 - A_2(\varepsilon_{n-1} + e_n) + o(\varepsilon_{n-1}))(\varepsilon_n + o(\varepsilon_n^2)) \\
&= A_2\varepsilon_{n-1}\varepsilon_n + o(\varepsilon_{n-1}\varepsilon_n) \\
&= A_2^2\varepsilon_{n-1}^2e_n + o(\varepsilon_{n-1}^2e_n).
\end{aligned} \tag{3.3.16}$$

Dividint les expresions de  $\varepsilon_n$  i  $e_{n+1}$  de (3.3.16) es té:

$$\frac{\varepsilon_n}{e_{n+1}} = \frac{A_2e_n\varepsilon_{n-1}}{A_2\varepsilon_n\varepsilon_{n-1}} = \frac{e_n}{\varepsilon_n} \Rightarrow \varepsilon_n^2 = e_ne_{n+1} \Rightarrow \varepsilon_{n-1}^2 = e_{n-1}e_n.$$

Per tant,

$$\begin{aligned}
e_{n+1}^{(16)} &= A_2^2\varepsilon_{n-1}^2e_n + o(e_n\varepsilon_{n-1}^2) \\
&= A_2e_n^2e_{n-1} + o(e_n^2e_{n-1}).
\end{aligned} \tag{3.3.17}$$

L'equació associada de (3.3.17) és  $t^2 - 2t - 1 = 0$ ; per tant, l'ordre de convergència és  $\rho_{16} = 1 + \sqrt{2}$ .

**TM:** De

$$\Phi_{17} = \begin{cases} y_n &= x_n - [x_{n-1}, y_{n-1}]_f^{-1} f(x_n), \\ x_{n+1} &= x_n - [x_n, y_n]_f^{-1} f(x_n), \end{cases}$$

es dedueix

$$\begin{aligned}
\varepsilon_n^{(17)} &= e_n - (1 - A_2(e_{n-1} + \varepsilon_{n-1}) + o(e_{n-1}))(e_n + \mathcal{O}(e_n^2)) \\
&= A_2e_{n-1}e_n \\
e_{n+1}^{(17)} &= e_n(1 - A_2(e_n + \varepsilon_n) + o(e_n))(e_n + A_2e_n^2 + \mathcal{O}(e_n^3)) \\
&= A_2\varepsilon_n e_n + o(\varepsilon_n e_n) \\
&= A_2e_n^2e_{n-1} + o(e_n^2e_{n-1}).
\end{aligned} \tag{3.3.18}$$

L'equació en diferències de  $e_{n+1}$  de (3.3.18) té l'equació associada  $t^2 - 2t - 1 = 0$ ; per tant  $\rho_{16} = 1 + \sqrt{2}$ .

## Capítol 4

# Eficiències

### 4.1 Estudi de les eficiències

Veiem a la taula 4.1 tots els mètodes iteratius presentats, amb l'ordre local de convergència i el cost computacional.

Per a cada un dels mètodes s'ha calculat el cost computacional que s'ha definit a (2.4.1). És a dir, s'han tingut en consideració el número d'evaluacions de funcions per iteració  $a$  i el número de productes i quocients que cal fer en cada iteració  $b$  i  $c$  respectivament.

Nom	Notació	$\rho$	$\mathcal{C}$
SM	$\phi_1$	$(1 + \sqrt{5})/2 = 1.61803$	$1\mu + 1 + l$
KM	$\phi_2$	2	$2\mu + 1 + l$
StM	$\phi_3$	2	$2\mu + 1 + l$
SKM	$\phi_4 = \phi_2 \diamond \phi_1$	$1 + \sqrt{3} = 2.732$	$3\mu + 2 + 2l$
SStM	$\phi_5 = \phi_3 \diamond \phi_1$	$1 + \sqrt{3} = 2.732$	$3\mu + 2 + 2l$
KSM	$\phi_6 = \phi_1 \diamond \phi_2$	$1 + \sqrt{3} = 2.732$	$3\mu + 2 + 2l$
KStM	$\phi_7 = \phi_3 \diamond \phi_2$	$1 + \sqrt{5} = 3.236$	$4\mu + 2 + 2l$
StSM	$\phi_8 = \phi_1 \diamond \phi_3$	3	$3\mu + 2 + 2l$
StKM	$\phi_9 = \phi_2 \diamond \phi_3$	4	$4\mu + 2 + 2l$
FSM	$\phi_{10}$	2	$2\mu + 2 + l$
FKM	$\phi_{11}$	$(1 + \sqrt{17})/2 = 2.562$	$3\mu + 2 + l$
FStM	$\phi_{12}$	3	$3\mu + 2 + l$
SSM	$\phi_{13} = \phi_1 \diamond \phi_1$	$1 + \sqrt{2} = 2.41421$	$2\mu + 2 + 2l$
KKM	$\phi_{14} = \phi_2 \diamond \phi_2$	$1 + \sqrt{5} = 3.236$	$4\mu + 2 + 2l$
StStM	$\phi_{15} = \phi_3 \diamond \phi_3$	4	$4\mu + 2 + 2l$
ShM	$\phi_{16}$	$1 + \sqrt{2} = 2.41421$	$2\mu + 2 + l$
TM	$\phi_{17}$	$1 + \sqrt{2} = 2.41421$	$2\mu + 2 + l$

Taula 4.1: Taula resum dels mètodes iteratius amb ordre de convergència i cost computacional.

Si s'ordenen per ordre de convergència, es veu a la taula 4.2 els resultats ordenats.

Nom	Notació	$\rho$	$\mathcal{C}$
SM	$\phi_1$	$(1 + \sqrt{5})/2 = 1.61803$	$1\mu + 1 + l$
KM, StM	$\phi_2, \phi_3$	2	$2\mu + 1 + l$
FSM	$\phi_{10}$	2	$2\mu + 2 + l$
ShM, TM	$\phi_{16}, \phi_{17}$	$1 + \sqrt{2} = 2.41421$	$2\mu + 2 + l$
SSM	$\phi_{13}$	$1 + \sqrt{2} = 2.41421$	$2\mu + 2 + 2l$
FKM	$\phi_{11}$	$(1 + \sqrt{17})/2 = 2.562$	$3\mu + 2 + l$
SKM, SStM, KSM	$\phi_4, \phi_5, \phi_6$	$1 + \sqrt{3} = 2.732$	$3\mu + 2 + 2l$
StSM	$\phi_8$	3	$3\mu + 2 + 2l$
FStM	$\phi_{12}$	3	$3\mu + 2 + l$
KStM, KKM	$\phi_7, \phi_{14}$	$1 + \sqrt{5} = 3.236$	$4\mu + 2 + 2l$
StKM, StStM	$\phi_9, \phi_{15}$	4	$4\mu + 2 + 2l$

Taula 4.2: Taula resum dels mètodes iteratius amb ordre de convergència i cost computacional ordenats per ordre.

A la taula 4.2 es mostren ordenats per ordre de convergència els mètodes presentats. Dins de cada ordre es poden veure quins mètodes tenen un cost computacional menor.

Es veu que  $\phi_9$  i  $\phi_{15}$  tenen la mateixa eficiència que  $\phi_2$  i  $\phi_3$  ja que tenen doble cost i ordre al quadrat del que tenia. És a dir, es pot simplificar com:

$$(\rho^2)^{1/2\mathcal{C}} = \rho^{1/\mathcal{C}}.$$

Aquesta situació i d'altres es reflecteixen en les següents observacions:

- $\phi_9$  i  $\phi_{15}$  estan en la mateixa posició que  $\phi_2$  i  $\phi_3$  a causa de la simplificació esmentada.
- Es veu que  $\phi_2$  i  $\phi_3$  és més eficient que  $\phi_{10}$  encara que té un ordre de convergència igual, ja que el cost computacional és menor. Té un quocient menys.
- $\phi_{12}$  és més eficient que  $\phi_8, \phi_6, \phi_5, \phi_4$  i  $\phi_{11}$ . Ja que té ordre igual o superior i, en canvi, cost computacional menor.
- $\phi_{16}$  i  $\phi_{17}$  són més eficients que  $\phi_{10}$  i  $\phi_{13}$ . Tenen ordre igual o superior i, en canvi, cost computacional menor. En el primer cas, malgrat tenir el mateix ordre  $\phi_{13}$  és més costós; per a  $\phi_{10}$  malgrat tenir el mateix cost computacional, té un ordre més petit.
- $\phi_9$  i  $\phi_{15}$  són més eficients que  $\phi_7$  i  $\phi_{14}$  perquè malgrat tenir el mateix cost, aquests MI són d'ordre més petit.

A la taula 4.3 es presenta una nova notació per caracteritzar els algorismes iteratius més eficients.

Notació	Nom	$\rho$	$\mathcal{C}$
$\Phi_1$	SM	$(1 + \sqrt{5})/2$	$1\mu + 1 + l$
$\Phi_2$	KM, StM, (StKM i StStM)	2	$2\mu + 1 + l$
$\Phi_3$	ShM, TM	$1 + \sqrt{2}$	$2\mu + 2 + l$
$\Phi_4$	FStM	3	$3\mu + 2 + l$

Taula 4.3: Classificació analítica dels mètodes més eficients.

Fent servir el factor de temps  $TF_k$  (2.4.3), es fa un estudi de les eficiències dels mètodes presentats a la taula 4.3,  $\Phi_k$ ,  $1 \leq k \leq 4$ , per a diferents valors de  $\ell$  (l'equivalent d'un quocient en productes). Els colors de les figures 4.1, 4.2 i 4.3 són els següents: coral ( $\Phi_1$ , mètode de la Secant), blau ( $\Phi_3$ , mètode de Shakno), turquesa ( $\Phi_4$ , mètode de Steffensen congelat) i magenta ( $\Phi_2$ , mètode de Kurxatov).

Per a valors mitjans o grans de  $\mu$ ,  $\mu \geq 20$ , s'observa gràficament a la figura 4.1 que  $TF_1$  està per sota dels altres  $TF_s$ ; és a dir,  $\Phi_1$  és més eficient en aquest cas. També es pot assegurar que:  $TF_3 < TF_4 < TF_2$ ; per tant, es té

$$CEI_1 > CEI_3 > CEI_4 > CEI_2.$$

A la figura 4.2 es veu el comportament dels factors de temps per a valors de  $\mu$  més petits. Es recorda que el valor de  $\mu$  determina el cost d'una avaluació de la funció expressat en productes.

Es pot veure a la figura 4.2 (a) que per a  $\ell = 1$ , i  $0 < \mu \leq \mu_1 = 3.9372$ ,  $\Phi_3$  és més eficient que  $\Phi_1$ ; mentre que per a  $\mu > \mu_1$  s'acompleix que  $CEI_1 > CEI_3$ .

En el segon cas (figura 4.2 (b)), per a  $\ell = 1.75$ , es té la mateixa situació d'abans per a  $\mu_2 = 7.6401$ ; és a dir,  $CEI_3 > CEI_1$  només si  $0 < \mu \leq \mu_2$ .

Per a  $\ell = 2.5$  i  $\mu_3 = 11.3430$ , s'obté  $CEI_3 > CEI_1$  si  $0 < \mu \leq \mu_3$ .

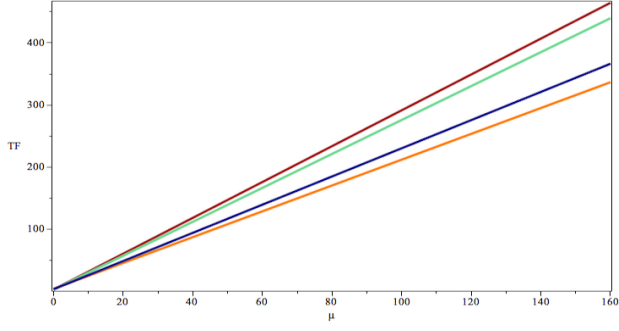
En altres paraules, quan  $\mu \geq 11.343$ , i tenint en compte que la funció  $TF_k$  és continua i creixent respecte l, podem escriure el següent teorema.

$$TF_k = \frac{C_k}{\log \rho_k} = \frac{a_k \mu + p_k + q_k \ell}{\log \rho_k}$$

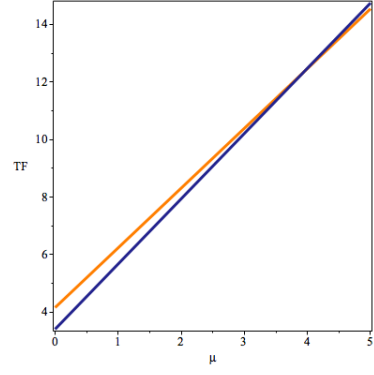
**Teorema 1.**  $\forall \ell \in [1.0, 2.5]$  i per a  $\mu$  més gran que 11.343 es té

$$CEI_1 > CEI_3 > CEI_4 > CEI_2.$$

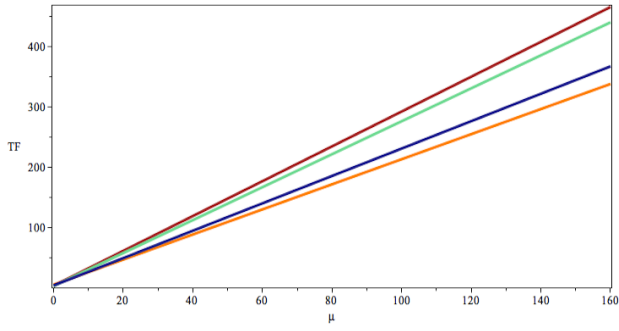
En particular el teorema és cert per les parelles  $(\ell, \mu_k)$ ,  $1 \leq k \leq 3$ , amb  $\mu_k$  valor frontera per sobre del qual  $CEI_1 > CEI_3$  següents:  $(1, 3.9372)$ ,  $(1.75, 7.6401)$  i  $(2.5, 11.343)$ .



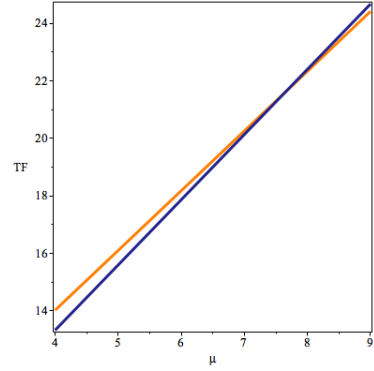
(a) Per a  $\ell = 1.0$



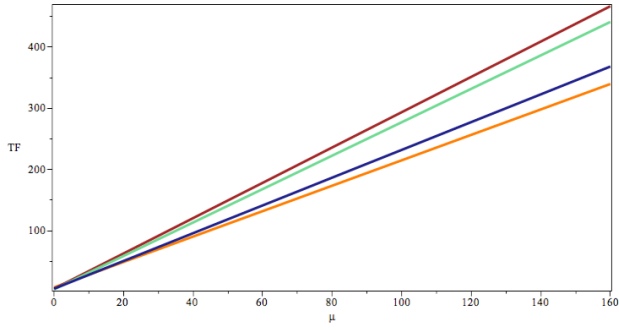
(a) Per a  $\ell = 1.0$ , Tall de  $\Phi_1$  amb  $\Phi_3$  a  $\mu = 3.937$



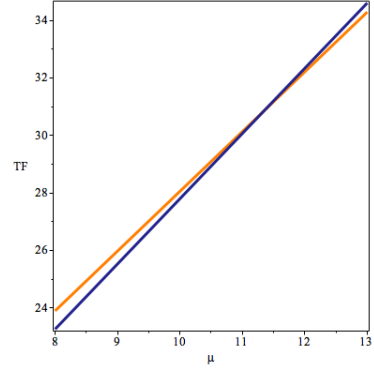
(b) Per a  $\ell = 1.75$



(b) Per a  $\ell = 1.75$ , Tall de  $\Phi_1$  amb  $\Phi_3$  a  $\mu = 7.640$



(c) Per a  $\ell = 2.5$



(c) Per a  $\ell = 2.5$ , Tall de  $\Phi_1$  amb  $\Phi_3$  a  $\mu = 11.343$

Figura 4.1:  $TF$  per a  $0 \leq \mu \leq 160$ .

Figura 4.2: Detalls de  $\mu$  en el tall de  $\Phi_1$  i  $\Phi_3$ .

A la figura 4.3 i tenint en compte el teorema anterior s'observa que, en general,  $\Phi_1$  és més eficient que  $\Phi_3$ ,  $\Phi_4$  i  $\Phi_2$ . A partir de (2.4.2), l'ordre és el següent:

$$CEI(\text{SM}) > CEI(\text{TM}) = CEI(\text{ShM}) > CEI(\text{FStM}) > CEI(\text{KM}) = CEI(\text{StM}) = CEI(\text{StKM})$$

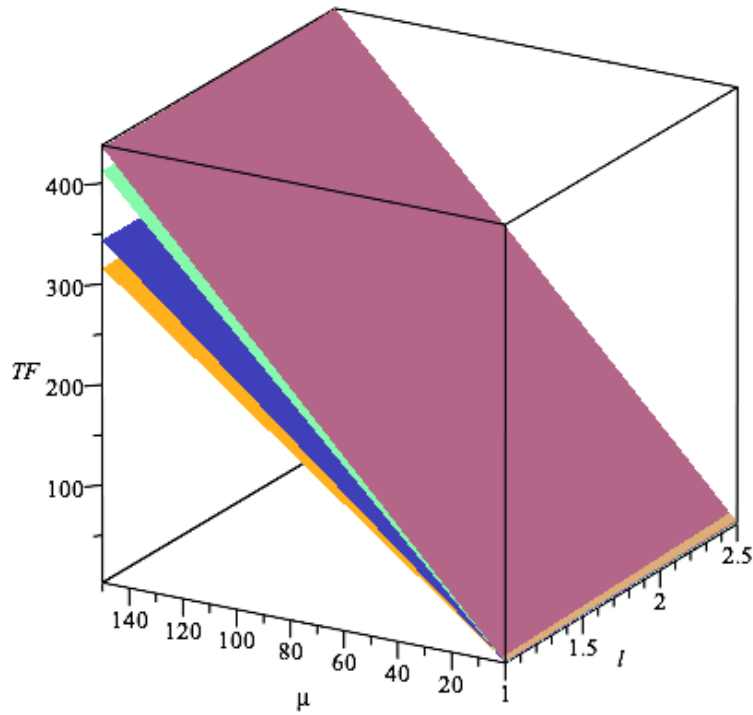


Figura 4.3:  $TF$  per als mètodes de la Secant (coral), Kursatov (magenta), Steffensen congelat (turquesa) i Shakno (blau).

Fent servir (2.4.2)

$$R_{i,j} = \frac{\log CEI_i}{\log CEI_j} = \frac{\log \rho_i}{\log \rho_j} \frac{\mathcal{C}_j}{\mathcal{C}_i}$$

per comparar els  $CEI$ 's es veu a la figura 4.4 la relació entre  $\mu$  i  $\ell$  i l'eficiència dels mètodes. La regió per sobre de la recta determina per quins valors de  $\mu$  i  $\ell$  el mètode  $\Phi_a$  és més eficient que l'algorisme  $\Phi_b$ . Per sota de la recta el mètode  $\Phi_b$  és més eficient.

## 4.2 Radi de convergència. Estudi Local

Per tal d'estudiar la grandària dels intervals de convergència des d'un punt de vista teòric suposem que:

**Hipòtesi 1.**

$$|f'(\alpha)^{-1}([x, y] - [u, v])| \leq K(|x - u| + |y - v|), \quad \forall x, y, u, v \in B_r(\alpha), \quad (4.2.1)$$

on  $B_r(\alpha) = \{x \in \mathbb{R} \mid |x - \alpha| < r\}$ ,  $\alpha$  és una arrel simple de  $f(x)$  i s'ha suposat  $f$  derivable en  $\alpha$ .

Es recorda el lema de Banach que es farà servir en els càlculs següents.

El lema de Banach [24] en el cas 1-dimensional afirma que si:

$$|1 - \mathcal{A}| < m < 1 \Rightarrow \mathcal{A}^{-1} < \frac{1}{1 - m}. \quad (4.2.2)$$

A partir de la definició de diferències dividides es té l'expressió:

$$f(x) - f(\alpha) = [x, \alpha](x - \alpha), \quad (4.2.3)$$

i aplicant (4.2.2) a  $\mathcal{A} = f'(\alpha)^{-1}[u, v]_f$ , és a dir,  $\mathcal{A}^{-1} = [u, v]_f^{-1}f'(\alpha)$ , es té:

$$\begin{aligned} |1 - \mathcal{A}| &= |1 - f'(\alpha)^{-1}[u, v]_f| \\ &= |f'(\alpha)^{-1}([\alpha, \alpha] - [u, v])|, \end{aligned} \quad (4.2.4)$$

on s'ha pres  $[\alpha, \alpha] = f'(\alpha)$ . Per la hipòtesi (4.2.1) i definint  $\varepsilon_u = u - \alpha$  i  $\varepsilon_v = v - \alpha$ , s'acompleix

$$\begin{aligned} |1 - \mathcal{A}| &\leq K(|u - \alpha| + |v - \alpha|) \\ &= K(|\varepsilon_u| + |\varepsilon_v|) \leq K 2r, \end{aligned} \quad (4.2.5)$$

ja que  $u, v \in B_r(\alpha)$ . Llavors, si  $K 2r < 1$ , aplicant (4.2.2) s'obté

$$|\mathcal{A}^{-1}| = |[u, v]_f^{-1}f'(\alpha)| \leq \frac{1}{1 - K(|\varepsilon_u| + |\varepsilon_v|)}. \quad (4.2.6)$$

### 4.2.1 Tècnica genèrica

Si es té un mètode iteratiu del tipus:  $y_n = x_n - [u, v]_f^{-1}f(x_n)$ , llavors,

$$y_n - \alpha = x_n - \alpha - [u, v]_f^{-1}(f(x_n) - f(\alpha)). \quad (4.2.7)$$

Tenint en consideració (4.2.3), es té

$$\begin{aligned} y_n - \alpha &= x_n - \alpha - [u, v]_f^{-1}[\alpha, x_n](x_n - \alpha) \\ &= (1 - [u, v]_f^{-1}[\alpha, x_n])(x_n - \alpha) \\ &= [u, v]_f^{-1}f'(\alpha) \{f'(\alpha)^{-1}([u, v]_f - [\alpha, x_n])\} (x_n - \alpha). \end{aligned} \quad (4.2.8)$$



Finalment, aplicant (4.2.6) i (4.2.1), de l'anterior equació (4.2.8) s'obté

$$|y_n - \alpha| \leq \frac{1}{1 - K(|\varepsilon_u| + |\varepsilon_v|)} K(|\varepsilon_u| + |v - x_n|) |x_n - \alpha|. \quad (4.2.9)$$

Com per a  $v \neq x_n$ ,  $|v - x_n| \leq |v - \alpha| + |x_n - \alpha| = \varepsilon_v + e_n$ , es té

$$|y_n - \alpha| \leq \frac{K(|\varepsilon_u| + |\varepsilon_v| + |e_n|)}{1 - K(|\varepsilon_u| + |\varepsilon_v|)} |x_n - \alpha|.$$

Si s'imposa que  $\frac{K(|\varepsilon_u| + |\varepsilon_v| + |e_n|)}{1 - K(|\varepsilon_u| + |\varepsilon_v|)} \leq 1$ , com  $|\varepsilon_j| < r$  amb  $j = u, v$  i  $|e_n| < r$ , llavors,

$$\frac{3rK}{1 - 2rK} \leq 1 \Rightarrow 3rK \leq 1 - 2rK \Rightarrow 5rK \leq 1.$$

En cas contrari si  $v = x_n$ , tot aplicant (4.2.9) queda

$$|y_n - \alpha| \leq \frac{K(|\varepsilon_u|)}{1 - K(|\varepsilon_u| + |\varepsilon_v|)} |x_n - \alpha|,$$

i, llavors,

$$\frac{Kr}{1 - 2rK} \leq 1 \Rightarrow 3rK \leq 1.$$

S'obté com a resultat que el radi de convergència local serà:

$$\begin{cases} r \leq \frac{1}{5K} & \text{si } v \neq x_n, \\ r \leq \frac{1}{3K} & \text{si } v = x_n. \end{cases} \quad (4.2.10)$$

Es procedeix a continuació aplicant aquesta tècnica als MI  $\Phi_k$ ,  $1 \leq k \leq 4$ .

## 4.2.2 Mètode de la Secant

Es recorda que el mètode iteratiu de la Secant consisteix en el següent algorisme:

$$\Phi_1(x_n; x_{n-1}) = x_n - [x_{n-1}, x_n]_f^{-1} f(x_n), \quad n \geq 0.$$

Se suposa que  $x_{n-1}, x_n \in B_r(\alpha)$ ; llavors, fent servir la mateixa tècnica que a l'apartat anterior

$$\begin{aligned} x_{n+1} - \alpha &= x_n - \alpha - [x_{n-1}, x_n]_f^{-1} (f(x_n) - f(\alpha)) \\ &= (1 - [x_{n-1}, x_n]_f^{-1} [\alpha, x_n]) (x_n - \alpha) \\ &= [x_{n-1}, x_n]_f^{-1} f'(\alpha) \{f'(\alpha)^{-1} ([x_{n-1}, x_n]_f - [\alpha, x_n])\} (x_n - \alpha). \end{aligned} \quad (4.2.11)$$

Per tant, es compleix:

$$|x_{n+1} - \alpha| \leq \frac{K|e_{n-1}|}{1 - K(|e_{n-1}| + |e_n|)} |x_n - \alpha|.$$

Com  $|e_j| < r$ , per  $n - 1 \leq j \leq n$ ,

si  $\frac{Kr}{1 - 2Kr} \leq 1 \Leftrightarrow r \leq \frac{1}{3K}$  i, llavors,  $x_{n+1} \in B_r(\alpha)$ .

### 4.2.3 Mètode de Shakno

Se suposa que  $y_{n-1}, x_n \in B_r(\alpha)$ . Una iteració del mètode iteratiu de Shakno té dos passos i ve donat per:

- Primer pas:  $y_n = y_{n-1} - [y_{n-1}, x_n]_f^{-1} f(y_{n-1})$ .

Llavors, utilitzant els resultats de la tècnica genèrica 4.2.1, es té:

$$\begin{aligned} y_n - \alpha &= y_{n-1} - \alpha - [y_{n-1}, x_n]_f^{-1} (f(y_{n-1}) - f(\alpha)) \\ &= (1 - [y_{n-1}, x_n]_f^{-1} [y_{n-1}, \alpha]_f) (y_{n-1} - \alpha) \\ &= [y_{n-1}, x_n]_f^{-1} f'(\alpha) \{ f'(\alpha)^{-1} ([y_{n-1}, x_n]_f - [y_{n-1}, \alpha]) \} (y_{n-1} - \alpha). \end{aligned} \quad (4.2.12)$$

Per tant, s'observa que

$$|y_n - \alpha| \leq \frac{K|e_n|}{1 - K(|\varepsilon_{n-1}| + |e_n|)} |y_{n-1} - \alpha|,$$

i com  $|\varepsilon_{n-1}| < r$  i  $|e_n| < r$ , s'obté que  $y_n \in B_r(\alpha)$  si  $r \leq \frac{1}{3K}$ , ja que  $\frac{Kr}{1 - 2Kr} < 1$ .

- Segon pas:  $x_{n+1} = y_n - [y_{n-1}, x_n]_f^{-1} f(y_n)$ .

S'obté de la mateixa manera

$$|x_{n+1} - \alpha| \leq \frac{K(|y_{n-1} - y_n| + |x_n - \alpha|)}{1 - K(|\varepsilon_{n-1}| + |e_n|)} |y_n - \alpha|. \quad (4.2.13)$$

De la desigualtat triangular següent  $|y_{n-1} - y_n| \leq |y_{n-1} - \alpha| + |y_n - \alpha| \leq 2r$ , s'obté a 4.2.13,  $\frac{3Kr}{1 - 2Kr} \leq 1 \Leftrightarrow r \leq \frac{1}{5K}$ .

### 4.2.4 Mètode de Traub

Se suposa que  $x_{n-1}, y_{n-1}, x_n \in B_r(\alpha)$ . Per tant,  $|\varepsilon_{n-1}| < r$  i  $|e_n| < r$ . Una iteració del mètode iteratiu de Traub té dos passos i ve donat per:

- Primer pas:  $y_n = x_n - [x_{n-1}, y_{n-1}]_f^{-1} f(x_n)$ .

Restant  $\alpha$  i seguint els mecanismes presentats anteriorment es té

$$\begin{aligned} y_n - \alpha &= x_n - \alpha - [x_{n-1}, y_{n-1}]_f^{-1} [x_n, \alpha]_f f(x_n - \alpha) \\ &= [x_{n-1}, y_{n-1}]_f^{-1} f'(\alpha) \{ f'(\alpha)^{-1} ([x_{n-1}, y_{n-1}]_f - [x_n, \alpha]) \} (x_n - \alpha). \end{aligned} \quad (4.2.14)$$

En definitiva,

$$|y_n - \alpha| \leq \frac{K(|x_n - x_{n-1}| + |y_{n-1} - \alpha|)}{1 - K(|\varepsilon_{n-1}| + |e_{n-1}|)} |x_n - \alpha|.$$

De  $|x_n - x_{n-1}| \leq |x_n - \alpha| + |x_{n-1} - \alpha| < 2r$ , s'obté

$$|y_n - \alpha| \leq \frac{K3r}{1 - K2r}.$$

Per tant,  $\frac{3rK}{1 - 2Kr} \leq 1 \Leftrightarrow r \leq \frac{1}{5K}$ .

- Segon pas:  $x_{n+1} = x_n - [x_n, y_n]_f^{-1} f(x_n)$ .

Tornant a aplicar el ja fet en el primer pas, s'obté:

$$x_{n+1} - \alpha = [x_n, y_n]_f^{-1} f'(\alpha) \{f'(\alpha)^{-1}([x_n, y_n]_f - [x_n, \alpha]_f)\} (x_n - \alpha). \quad (4.2.15)$$

De (4.2.15)

$$|x_{n+1} - \alpha| \leq \frac{K|y_n - \alpha|}{1 - K(|e_n| + |\varepsilon_n|)} |x_n - \alpha|,$$

$$\frac{Kr}{1 - 2Kr} \leq 1 \Leftrightarrow r \leq \frac{1}{3K} \text{ malgrat en el primer pas és } \frac{1}{5K}.$$

#### 4.2.5 Mètode de Kurxatov

El mètode iteratiu de Kurxatov consisteix en el següent esquema iteratiu d'un pas:

$$\phi_2(x_n; x_{n-1}) = x_n - [x_{n-1}, 2x_n - x_{n-1}]_f^{-1} f(x_n) \quad n \geq 0.$$

Se suposa que  $x_{n-1}, x_n \in B_r(\alpha)$ ; llavors, fent servir la mateixa tècnica que a l'apartat anterior

$$x_{n+1} - \alpha = x_n - \alpha - [x_{n-1}, 2x_n - x_{n-1}]_f^{-1} (f(x_n) - f(\alpha)) \quad (4.2.16)$$

$$e_{n+1} = [x_{n-1}, 2x_n - x_{n-1}]_f^{-1} f'(\alpha) \{f'(\alpha)^{-1}([x_{n-1}, 2x_n - x_{n-1}] - [\alpha, x_n])\} (x_n - \alpha) \quad (4.2.17)$$

**Hipòtesi 2.** Suposant que  $\exists L > 0$  tal que  $\forall x_{n-1}, x_n \in J_\alpha$  es té

$$|f'(\alpha)^{-1}([x_{n-1}, 2x_n - x_{n-1}] - [x_n, x_n])| \leq L|x_n - x_{n-1}|^2. \quad (4.2.18)$$

A partir de (4.2.18) i (4.2.1), es té

$$f'(\alpha)^{-1}([x_{n-1}, 2x_n - x_{n-1}] - [x_n, x_n]) + [x_n, x_n] - [\alpha, x_n] \leq L|x_n - x_{n-1}|^2 + K|x_n - \alpha| \quad (4.2.19)$$

Per tant, es compleix:

$$|e_{n+1} - \alpha| \leq \frac{K|x_n - \alpha| + L|x_n - x_{n-1}|^2}{1 - 2K|x_n - \alpha| - L|x_n - x_{n-1}|^2} |x_n - \alpha|.$$

De  $x_{n-1}, x_n \in B_r(\alpha)$ , si  $|x_n - x_{n-1}| \leq |x_n - \alpha| + |x_{n-1} - \alpha| \leq 2r$ , llavors,  $x_{n+1} \in B_r(\alpha)$  si s'acompleix

$$\frac{Kr + 4Lr^2}{1 - 2Kr - 4Lr^2} r < r \Leftrightarrow r \leq \frac{2}{3K + \sqrt{9K + 32L}}.$$

### 4.2.6 Mètode congelat de Steffensen

Es considera que  $x_0 \in B_r(\alpha)$ . Una iteració de l'algorisme iteratiu congelat de Steffensen de dos passos ve donat per:

- Primer pas:  $y_n = x_n - [x_n, x_n + f(x_n)]_f^{-1} f(x_n)$ .

Restant  $\alpha$  a tos dos costats de la igualtat i si es defineix  $\varepsilon_n = y_n - \alpha$ , es té

$$\begin{aligned}\varepsilon_n &= x_n - \alpha - [x_n, x_n + f(x_n)]_f^{-1} (f(x_n) - f(\alpha)) \\ &= (1 - [x_n, x_n + f(x_n)]_f^{-1} [x_n, \alpha]) (x_n - \alpha) \\ &= [x_n, x_n + f(x_n)]_f^{-1} f'(\alpha) \{f'(\alpha) ([x_n, x_n + f(x_n)] - [x_n, \alpha])\} (x_n - \alpha).\end{aligned}\tag{4.2.20}$$

**Hipòtesi 3.** Si se suposa que  $\exists M > 0$  tal que  $\forall x \in J_\alpha$

$$|f'(x)| < M.\tag{4.2.21}$$

Tot aplicant (4.2.21), s'obté

$$\begin{aligned}|f'(\alpha)^{-1}([\alpha, \alpha] - [x_n, x_n + f(x_n)])| &\leq K(2|x_n - \alpha| + |f(x_n)|) \\ &\leq K(2|x_n - \alpha| + |M||x_n - \alpha|) \\ &\leq (2 + M)|x_n - \alpha| < 1.\end{aligned}\tag{4.2.22}$$

Per tant,  $|\varepsilon_n| \leq \frac{1}{1 - K(2 + M)|e_n|} K(1 + M)|x_n - \alpha|^2$ .

- Segon pas:  $x_{n+1} = y_n - [x_n, x_n + f(x_n)]_f^{-1} f(y_n)$ .

Restant  $\alpha$  dels dos membres de l'equació anterior s'obté

$$e_{n+1} = [x_n, x_n + f(x_n)]_f^{-1} f'(\alpha) \{f'(\alpha) - ([x_n, x_n + f(x_n)] - [\alpha, y_n])\} (y_n - \alpha).\tag{4.2.23}$$

Aplicant (4.2.21) es té

$$|e_{n+1}| \leq \frac{K((2 + M)|e_n| + |\varepsilon_n|)}{1 - K(2 + M)|e_n|} |\varepsilon_n| < r,$$

i recordant el resultat obtingut pel primer pas de l'algorisme, es té

$$|\varepsilon_n| \leq \frac{K(1 + M)}{1 - K(2 + M)|e_n|} |e_n|^2 < r.$$

Com  $|e_n| < r$  i  $|\varepsilon_n| < r$ , de  $|e_{n+1}| < r$ , es té

$$\frac{K(3 + M)r}{1 - K(2 + M)r} r < r \Leftrightarrow r \leq \frac{1}{K(5 + 2M)}.$$

### 4.3 Temps d'execució d'operacions bàsiques i funcions elementals

A les taules 4.4 i 4.5 es presenta el cost computacional de diferents funcions elementals en relació al cost del producte  $x * y$ , on  $x = \sqrt{3} - 1$ ,  $y = \sqrt{5}$  i la constant  $k = \sqrt{2}$  es pren de doble precisió. El programa que computa les funcions està escrit en C++ amb les llibreries libgmp i libmpfr [12] en una màquina Intel Core i7, 2.7 Ghz.

Aritmètica	$x * y$	$x * x$	$x/y$	$e^x$	$x^x$	$\ln x$	$\log x$	$\sin x$	$\cos x$	$\tan x$
4096 <i>digits</i>	0.011795 ms 1.000000	0.007389 ms 0.626453	0.021149 ms <b>1.793028</b>	1.034213 ms 87.681336	1.884653 ms 159.782245	0.746458 ms 63.285270	1.533724 ms 130.030261	1.217986 ms 103.261762	1.175953 ms 99.698154	1.246334 ms 105.665126
2048	0.004231 ms	0.598286	1.672678	68.229202	134.413180	56.821868	115.590716	72.904807	70.018631	76.368218
1024	0.001359 ms	0.663508	1.821756	63.914015	134.393821	59.153028	122.176201	61.308947	55.335254	62.746226
512	0.000523 ms	0.566606	1.937957	56.964346	134.898680	60.848543	125.204662	50.044535	43.533490	53.197867
256	0.000201 ms	0.659751	1.981032	65.517979	178.461856	74.395342	155.243605	54.857555	43.931569	60.472298
128	0.000087 ms	0.745566	2.263302	94.734687	259.845696	82.772480	181.961838	68.933410	54.483151	78.406843
64	0.000054 ms	0.874931	2.550306	118.110508	344.943552	97.476745	217.153401	79.617867	58.130181	94.701859
32	0.000037 ms	0.866667	1.490763	142.908171	414.541253	77.674041	209.325286	86.670042	64.102932	108.466274
16	0.000026 ms	0.861408	1.824225	114.659372	435.852333	241.289931	209.416007	106.366398	68.362881	124.394602

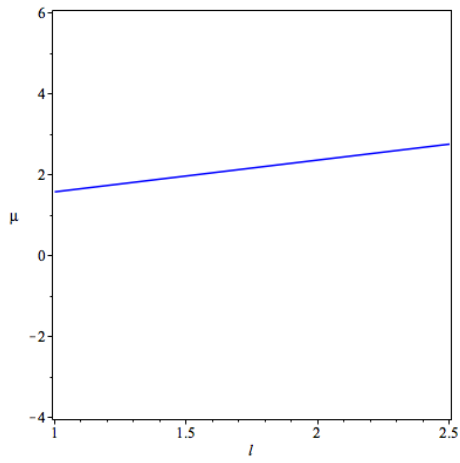
Taula 4.4: Cost computacional de les funcions elementals respecte el producte  $x * y$ .

Aritmètica	$x * y$	$t_n/t_{4096}$	$x * x$	$x * k$	$x/y$	$x/k$	$x + y$	$x + k$	$x - y$	$x - k$
4096	0.011795 ms 1.000000		0.007389 ms 0.626453	0.000321 ms 0.027207	0.021149 ms <b>1.793028</b>	0.001131 ms 0.095892	0.000188 ms 0.015898	0.000301 ms 0.025529	0.000266 ms 0.022538	0.000269 ms 0.022801
2048	0.004231 ms	0.358711	0.598286	0.057879	1.672678	0.161632	0.024600	0.051623	0.033927	0.051285
1024	0.001359 ms	0.115218	0.663508	0.146491	1.821756	0.307544	0.042873	0.130789	0.060263	0.135526
512	0.000523 ms	0.044341	0.566606	0.335310	1.937957	0.521898	0.073506	0.333941	0.098996	0.338732
256	0.000201 ms	0.017041	0.659751	0.787196	1.981032	1.118554	0.145599	0.747481	0.200356	0.770599
128	0.000087 ms	0.007376	0.745566	1.783083	2.263302	2.450205	0.303206	1.863574	0.410982	1.897681
64	0.000054 ms	0.004578	0.874931	3.132852	2.550306	3.117288	0.422735	2.910506	0.599222	2.810451
32	0.000037 ms	0.003137	0.866667	3.437751	1.490763	4.610442	0.659036	4.578314	0.821687	4.285944
16	0.000026 ms	0.002204	0.861408	6.070987	1.824225	6.395494	0.863662	6.368451	1.126197	5.980846

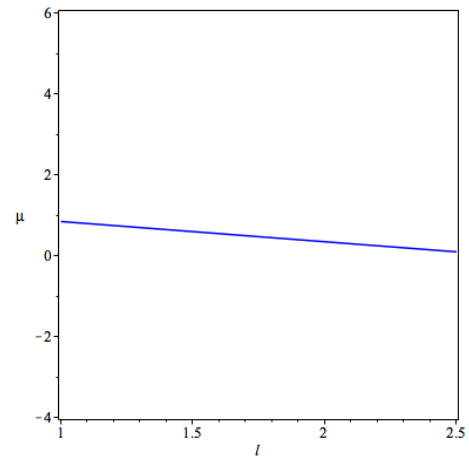
Taula 4.5: Cost computacional d'altres funcions elementals respecte el producte  $x * y$ .

Cal destacar algunes dades significatives de la taula 4.4. L'equivalent en productes d'un quocient  $\ell$  pren valors  $1.5207 < \ell \leq 2.5250$ . Per a una aritmètica fixa de 4096 dígitos es té  $\ell = 1.7930$ . L'estudi general fet a la secció 4.1 per a valors d' $\ell$   $1.0 \leq \ell \leq 2.5$  és més general tenint en compte altres màquines i programaris.

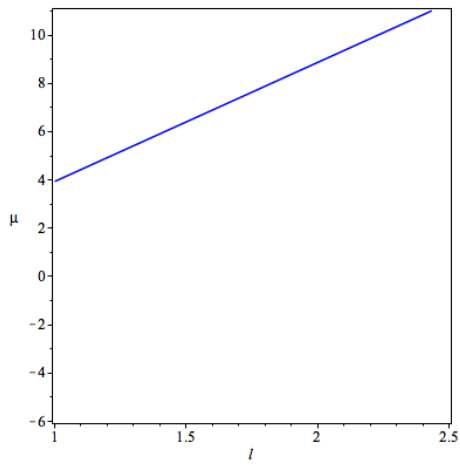
A la taula 4.5 es veu el cost d'algunes operacions on un dels operands és una constant. A la tercera columna es veu la relació del cost del producte per les diferents aritmètiques, on  $t_n$  és el temps en l'execució d'un producte amb  $n$  dígitos i, per tant,  $\frac{t_n}{t_{4096}}$  és el ratio dels temps d'execució d'un producte respecte l'aritmètica de 4096 dígitos.



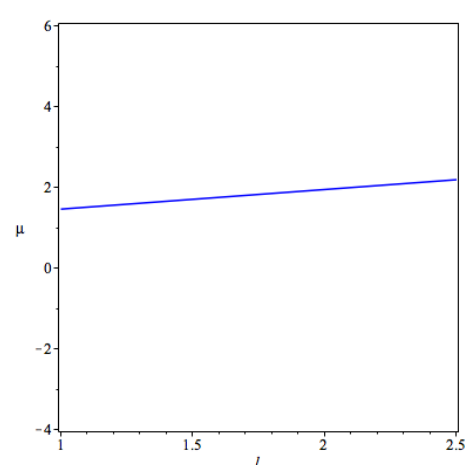
(a)  $\Phi_1$  i  $\Phi_2$



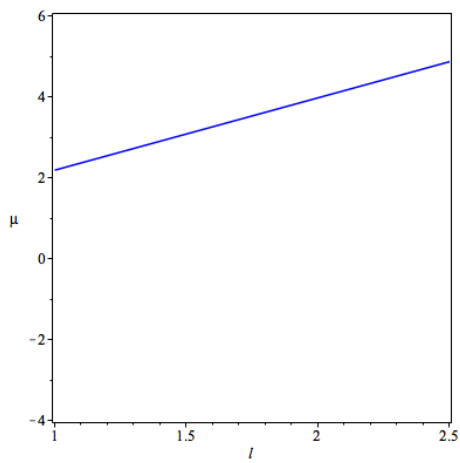
(d)  $\Phi_3$  i  $\Phi_2$



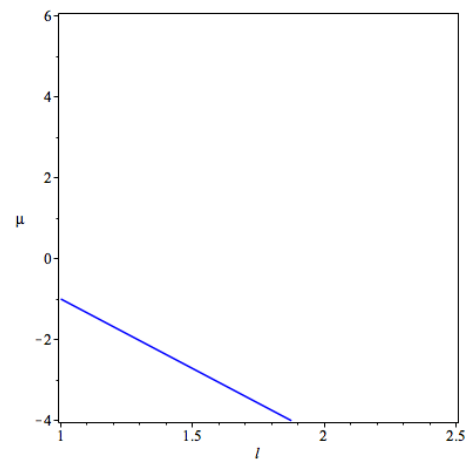
(b)  $\Phi_1$  i  $\Phi_3$



(e)  $\Phi_3$  i  $\Phi_4$



(c)  $\Phi_1$  i  $\Phi_4$



(f)  $\Phi_4$  i  $\Phi_2$

Figura 4.4: Comparació entre els *CEI*'s per a  $1 \leq l \leq 2.5$  i  $\mu$  dels mètodes  $\Phi_a$  i  $\Phi_b$

## Capítol 5

# Resultats numèrics

En aquesta secció es presenten els resultats numèrics obtinguts en la resolució de les 10 equacions no lineals,  $f_j(x)$  on  $1 \leq j \leq 10$ , presentades a la taula 5.1 prenent els mètodes iteratius  $\Phi_i$ ,  $1 \leq i \leq 4$ , presentats a la taula 4.3, com:

- Mètode de la Secant ( $\Phi_1$ , SM (2.2.1)),
- Mètode de Kurxatov ( $\Phi_2$ , KM (2.2.2)),
- Mètode de Shakno ( $\Phi_3$ , ShM (3.1.13)),
- Mètode de Steffensen congelat ( $\Phi_4$ , FStM (3.1.9)).

Per a cada funció es dona una aproximació inicial  $[x_{-1}, x_0]$  de l'interval on es troba l'arrel  $\alpha$ , una aproximació amb 16 dígitos significatius d' $\alpha$  i el cost d'una avaluació de la funció expressat en productes  $\mu$ .

Com a resultats numèrics es presenta la resolució de les funcions mostra amb el mètodes iteratius més eficients amb aritmètica fixa i adaptativa i amb tres programaris diferents. Es fa un estudi dels radis de convergència com també de les aproximacions inicials i el número de iteracions.

Tots els resultats numèrics presentats, excepte la comparativa entre MPFR, ARPREC i BigFloat, s'han calculat amb la llibreria de C++ MPFR de multiprecisió [12] amb 4096 dígitos de mantissa, en el cas d'aritmètica fixa, i amb una aritmètica entre 32 i 4096 dígitos de mantissa en el cas d'aritmètica adaptativa.

Tots els programes han estat compilats amb gcc amb les llibreries libgmp (6.0.0) i libmpfr (3.1.2) en un Intel Core i7, 2.7 Ghz amb 6 Mb de cache.

Per aquest hardware i software el cost computacional del quocient respecte el producte és  $\ell = 1.7930$ .

$f_i(x)$	$\alpha$	$x_{-1}$	$x_0$	$\mu$
$f_1(x) = x^x - 10$	2.506184145588769	3.0	2.5	159.807774
$f_2(x) = x \log(x) - 1$	1.763222834351896	1.0	2.0	64.308071
$f_3(x) = \tan(x) - \frac{1}{1+x^2}$	0.623899560580903	0.5	1.0	108.132674
$f_4(x) = x + \frac{1}{x} - e^x + 3$	1.660670179870772	2.0	1.0	89.538329
$f_5(x) = x^3 - 3x^2 + x - 2$	2.893289196304497	3.0	2.0	1.698999
$f_6(x) = e^{-x} + \cos(x)$	1.746139530408012	1.5	1.0	187.395388
$f_7(x) = \frac{1}{(x-0.3)^2+0.01} + \frac{1}{(x-0.9)^2+0.04} - 6.0$	1.299549682584821	1.0	1.29	5.713665
$f_8(x) = 16x^4 - 58x^3 + 79x^2 - 56x + 18$	0.875965279525090	1.0	0.8	2.821784
$f_9(x) = e^{x^2+7x-30} - 1$	3.000000000000000	3.10	2.995	88.396496
$f_{10}(x) = 2 \sin(x) - x + 1$	2.380061273139339	2.5	2.0	103.337036

Taula 5.1: Conjunt de funcions prova amb l'arrel  $\alpha$  aproximada a 16 decimals, els punts inicials d'iteració i el cost d'una avaluació de la funció per a 4096 dígits de precisió expressat en productes ( $\mu$ ).

Es poden veure els resultats de costos computacionals complets a la taula 4.4.

Tots quatre mètodes necessiten unes aproximacions inicials. Considerant  $x_{-1}, x_0$  per els esquemes  $\Phi_1$  i  $\Phi_2$ ,  $y_{-1} = x_{-1}, x_0$  per al mètode iteratiu de Shakno  $\Phi_3$  i  $x_0$  per a l'únic mètode que es considera sense memòria  $\Phi_4$ .

A partir de les aproximacions inicials descrites els mètodes iteratius anteriors donen lloc a una successió infinita  $\mathcal{S} = \{x_1, x_2, \dots, x_n, x_{n+1}, \dots\}$ , de la qual només ens cal un subconjunt finit, és a dir  $S = \{x_1, x_2, \dots, x_I\}$ .

El clàssic criteri de tall  $|x_I - \alpha| > 0.5 \times 10^{-\eta}$  i  $|x_{I+1}| \leq 0.5 \times 10^{-\eta}$  és reemplaçat per (2.6.4):

$$d_I < \eta \quad \text{i} \quad \rho d_I > \eta,$$

on  $\eta = 4096$  i  $\rho$  és l'ordre de convergència del mètode iteratiu.



Per cada ordre computacional de convergència explicat en la secció (2.5) tenim els diferents criteris de tall:

$$\begin{aligned}\hat{d}_I < \eta \quad \text{i} \quad \frac{\rho^2}{\rho - 1} \hat{d}_I > \eta, \\ \tilde{d}_I < \eta \quad \text{i} \quad \frac{\rho^2}{2\rho - 1} \tilde{d}_I > \eta, \\ \check{d}_I < \eta \quad \text{i} \quad \frac{\rho}{\rho - 1} \check{d}_I > \eta.\end{aligned}$$

Els ordres de convergència local (2.5.9 - 2.5.12) seran:  $\bar{\lambda} = \bar{\lambda}_I, \hat{\lambda} = \hat{\lambda}_I, \tilde{\lambda} = \tilde{\lambda}_I, \check{\lambda} = \check{\lambda}_I$ .

Amb la resolució de les funcions mostra  $f_j(x)$ ,  $1 \leq j \leq 10$ , presentades a la taula 5.1 amb els mètodes  $\Phi_i$ ,  $1 \leq i \leq 4$ , s'espera observar el comportament dels mètodes iteratius representat a la figura 5.1.

La figura 5.1a mostra per  $\ell = 1.793028$  els factors de temps TF dels algorismes iteratius  $\Phi_i$ ,  $1 \leq i \leq 4$  respecte al cost computacional  $\mu$ . En detall a les figures 5.1b i 5.1c el comportament de  $\Phi_1$ - $\Phi_3$  i  $\Phi_2$ - $\Phi_4$  respectivament.

Per les funcions amb  $\mu > 7.852568712$ , punt de talls dels mètodes  $\Phi_1$ - $\Phi_3$ , s'espera que el comportament dels mètodes ordenats per eficiència sigui  $\Phi_1 > \Phi_3 > \Phi_4 > \Phi_2$ , entenent que  $\Phi_1$  és el més eficient i  $\Phi_4$  el menys eficient.

Per les funcions amb  $\mu \leq 7.852568712$  s'espera un comportament diferents dels mètodes  $\Phi_1$  i  $\Phi_3$  donant la següent classificació respecte les eficiències  $\Phi_3 > \Phi_1 > \Phi_4 > \Phi_2$ .

Per a qualsevol  $\mu$  s'espera que  $\Phi_4$  sigui més eficient que  $\Phi_2$ .

## 5.1 Resultats amb Aritmètica Fixa

Les taules generades per les resolucions de les funcions test  $f_j(x)$  amb  $1 \leq j \leq 10$  amb els mètodes  $\Phi_i$  amb  $1 \leq i \leq 4$  es troben a l'annex A.1.

El número de decimals correctes s'aproxima i es representa en els resultats numèrics com  $D_n = \lceil -\log_{10} |e_n| \rceil$  per  $n = I$ . També es defineix  $\hat{D}_I$ ,  $\tilde{D}_I$  i  $\check{D}_I$ .

A la taula 5.2 es presenten les taules que contenen el número d'iteracions ( $I$ ), número de decimals correctes per als paràmetres d'error CLOC ( $D_I$ ), ACLOC ( $\hat{D}_I$ ), ECLOC ( $\tilde{D}_I$ ) i PCLOC ( $\check{D}_I$ ), índex d'eficiència computacional ( $CEI$ ), factor temps ( $TF$ ) i el temps d'execució ( $\tau$ ) dels mètodes iteratius  $\Phi_i$  amb  $1 \leq i \leq 4$  per a les funcions mostra  $f_j(x)$  amb  $1 \leq j \leq 10$  de la taula 5.1. Tots aquests càlculs s'han fet per a les mateixes condicions inicials.

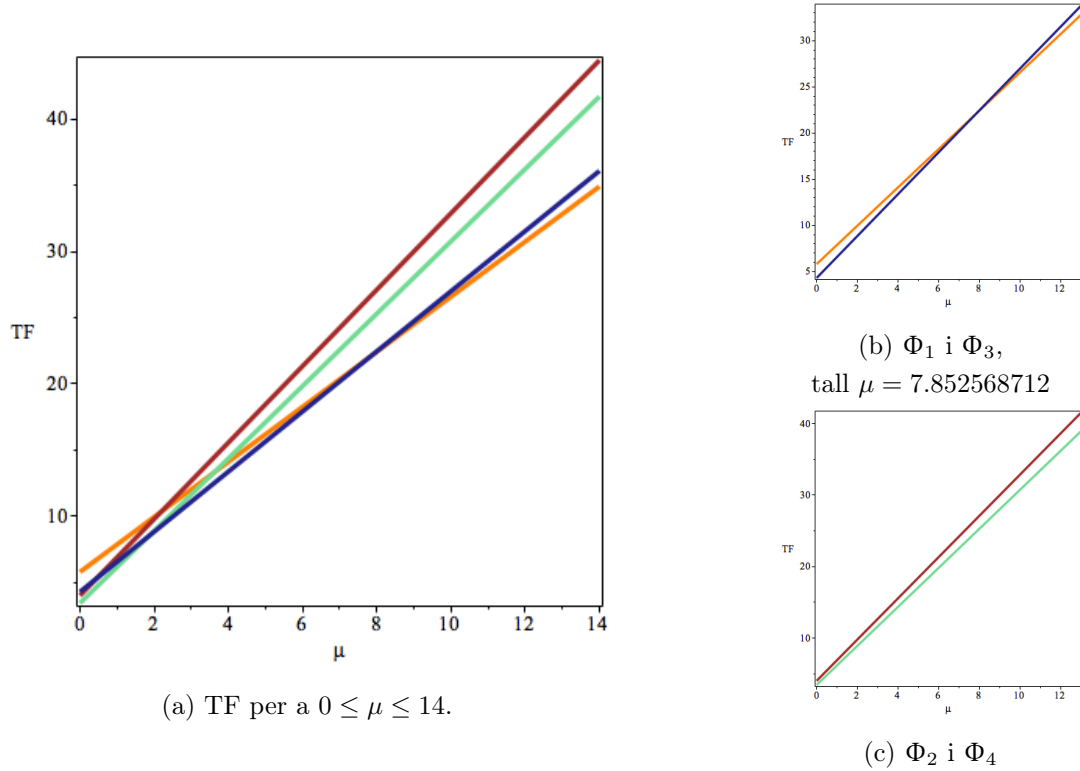


Figura 5.1: TF i detalls de  $\mu$  per a  $\Phi_i$   $0 \leq i \leq 4$  per a  $\ell = 1.7930$ .

Les taules de 5.3 mostren els resultats dels ordres locals de convergència ACLOC, CLOC, ECLOC i PCLOC per als mètodes iteratius  $\Phi_i$  amb  $1 \leq i \leq 4$  aplicats a les funcions  $f_j(x)$  amb  $1 \leq j \leq 10$  de la taula 5.1.

A la figura 5.2 es veuen les gràfiques de les rectes de regressió dels punts  $(\log(-\log_{10}|\hat{e}_n|), t_n)$ ,  $n = 1, \dots, I$  per als algorismes  $\Phi_i$  amb  $1 \leq i \leq 4$  aplicats a les funcions  $f_j(x)$  amb  $1 \leq j \leq 10$ . Els colors identifiquen els mètodes de la següent manera: coral per  $\Phi_1$ , magenta per  $\Phi_2$ , blau per  $\Phi_3$  i turquesa per  $\Phi_4$ .

Els pendents de les rectes de regressió indiquen l'eficiència dels mètodes [16]. A menys pendent més eficiència. Es poden veure els valors a la taula 5.4 .

S'observa per al conjunt de funcions mostra que s'han fet servir que, en general, el mètode més eficient és el mètode de la Secant (SM,  $\Phi_1$ ) seguit del mètode de Shakno (ShM,  $\Phi_3$ ). El comportament dels mètodes de Kurxatov (KM,  $\Phi_2$ ) i Steffensen Congelat (FStM,  $\Phi_4$ ) varia en la mostra de funcions.

Les funcions  $f_5(x)$ ,  $f_7(x)$  i  $f_8(x)$  mostren que el mètode Shakno (ShM,  $\Phi_3$ ) és més eficient que el mètode de la Secant (SM,  $\Phi_1$ ). Aquest fet és a causa de que el cost computacional  $\mu$  d'aquestes funcions és menor al punt de tall 7.852568712 (figura 5.1b) i fins a aquest punt  $\Phi_3$  és més eficient

$f_1(x), x_{-1} = 3.0, x_0 = 2.5$								
	$I$	$D_I$	$\hat{D}_I$	$\tilde{D}_I$	$\tilde{D}_n$	CEI	TF	$\tau$
$\Phi_1$	16	2718	2718	2718	2718	1.0030	337.96	64
$\Phi_2$	13	2057	2057	2057	2057	1.0022	465.18	87
$\Phi_3$	9	3699	3699	3699	3699	1.0027	367.90	63
$\Phi_4$	7	1871	1870	1872	1870	1.0023	440.61	64

$f_2(x), x_{-1} = 1.0, x_0 = 2.0$								
	$I$	$D_I$	$\hat{D}_I$	$\tilde{D}_I$	$\tilde{D}_n$	CEI	TF	$\tau$
$\Phi_1$	16	2942	2943	2942	2943	1.0072	139.50	67
$\Phi_2$	11	3200	3203	3201	3200	1.0053	189.62	53
$\Phi_3$	9	3490	3490	3489	3490	1.0066	151.19	46
$\Phi_4$	7	2101	2101	2101	2102	1.0055	179.83	44

$f_3(x), x_{-1} = 0.5, x_0 = 1.0$								
	$I$	$D_I$	$\hat{D}_I$	$\tilde{D}_I$	$\tilde{D}_n$	CEI	TF	$\tau$
$\Phi_1$	17	3654	3655	3655	3655	1.0044	230.57	81
$\Phi_2$	13	2480	2480	2480	2480	1.0032	316.07	78
$\Phi_3$	9	3040	3040	3040	3040	1.0041	250.64	62
$\Phi_4$	8	1797	1798	1796	1798	1.0034	299.50	65

$f_4(x), x_{-1} = 2.0, x_0 = 1.0$								
	$I$	$D_I$	$\hat{D}_I$	$\tilde{D}_I$	$\tilde{D}_n$	CEI	TF	$\tau$
$\Phi_1$	17	2713	2713	2713	2713	1.0052	191.93	80
$\Phi_2$	12	2326	2326	2326	2326	1.0038	262.42	70
$\Phi_3$	9	2483	2484	2483	2484	1.0048	208.44	54
$\Phi_4$	9	2820	2819	2820	2819	1.0040	248.73	67

$f_5(x), x_{-1} = 3.0, x_0 = 2.0$								
	$I$	$D_I$	$\hat{D}_I$	$\tilde{D}_I$	$\tilde{D}_n$	CEI	TF	$\tau$
$\Phi_1$	17	3080	3080	3080	3080	1.1092	9.39	55
$\Phi_2$	12	3948	3949	3948	3949	1.1131	8.97	42
$\Phi_3$	9	1925	1925	1925	1925	1.1122	9.12	30
$\Phi_4$	9	3127	3126	3128	3126	1.1147	8.86	37

$f_6(x), x_{-1} = 1.5, x_0 = 1.0$								
	$I$	$D_I$	$\hat{D}_I$	$\tilde{D}_I$	$\tilde{D}_n$	CEI	TF	$\tau$
$\Phi_1$	16	3336	3337	3337	3337	1.0025	395.29	91
$\Phi_2$	12	3109	3611	3108	3109	1.0018	544.78	98
$\Phi_3$	8	1853	2018	1853	1854	1.0023	430.50	69
$\Phi_4$	7	3424	3426	3423	3426	1.0019	515.95	80

$f_7(x), x_{-1} = 1.0, x_0 = 1.29$								
	$I$	$D_I$	$\hat{D}_I$	$\tilde{D}_I$	$\tilde{D}_n$	CEI	TF	$\tau$
$\Phi_1$	23	4021	4020	4020	4020	1.0704	17.73	74
$\Phi_2$	14	3802	3801	3802	3801	1.0632	20.56	50
$\Phi_3$	10	1438	2637	2189	2190	1.0694	18.23	44
$\Phi_4$	8	1741	1740	1743	1740	1.0652	19.83	27

$f_8(x), x_{-1} = 1.0, x_0 = 0.8$								
	$I$	$D_I$	$\hat{D}_I$	$\tilde{D}_I$	$\tilde{D}_n$	CEI	TF	$\tau$
$\Phi_1$	17	2584	2585	2583	2585	1.0960	11.73	55
$\Phi_2$	14	3234	3233	3234	3233	1.0944	12.21	57
$\Phi_3$	9	2135	2136	2134	2136	1.0971	11.67	30
$\Phi_4$	7	2146	2147	2146	2147	1.0964	11.93	27

$f_9(x), x_{-1} = 3.10, x_0 = 2.995$								
	$I$	$D_I$	$\hat{D}_I$	$\tilde{D}_I$	$\tilde{D}_n$	CEI	TF	$\tau$
$\Phi_1$	19	3356	3355	3293	3355	1.0053	189.56	89
$\Phi_2$	14	2661	2660	2661	2660	1.0039	259.13	79
$\Phi_3$	10	2486	2485	2486	2485	1.0049	205.85	60
$\Phi_4$	8	1794	1792	1796	1792	1.0041	245.61	51

$f_{10}(x), x_{-1} = 2.5, x_0 = 2.0$								
	$I$	$D_I$	$\hat{D}_I$	$\tilde{D}_I$	$\tilde{D}_n$	CEI	TF	$\tau$
$\Phi_1$	16	2862	2863	2862	2863	1.0046	220.60	79
$\Phi_2$	12	3981	3982	3981	3981	1.0033	302.24	75
$\Phi_3$	9	3920	3921	3920	3921	1.0042	239.76	63
$\Phi_4$	7	2838	2838	2837	2838	1.0035	286.41	57

Taula 5.2: Execucio de les 10 funcions mostra per als algorismes  $\Phi_i$  amb  $1 \leq i \leq 4$  amb aritmètica fixa.

que  $\Phi_1$  com mostra la figura 4.4c.

Per a les funcions  $f_4(x)$ ,  $f_8(x)$  i  $f_9(x)$ , encara que el pendent de la recta pels valors del mètode Kurxatov (KM,  $\Phi_2$ ) és menys pronunciada que la pendent de la recta pel mètode d'Steffensen Congelat (FStM,  $\Phi_4$ ), el número de dígitos correctes respecte el temps mitjà d'execució indica que el mètode  $\Phi_4$  és més eficient que  $\Phi_2$ .

S'ha de fer notar que en fer l'execució dels mètodes iteratius amb una aritmètica amb més bits de mantissa (32784 dígitos), veure a la taula 5.5, els resultats trobats teòricament es compleixen en mantisses més grans. L'execució dels mètodes iteratius fent servir 4096 dígitos no té un nombre

CLOC										
	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$
$\Phi_1$	1.618	1.619	1.618	1.618	1.618	1.619	1.618	1.62	1.617	1.619
$\Phi_2$	1.999	2.002	2.000	2.000	2.000	2.003	1.999	1.999	1.997	2.002
$\Phi_3$	2.414	2.418	2.416	2.417	2.416	2.423	2.418	2.429	2.408	2.417
$\Phi_4$	2.962	3.007	3.024	2.996	2.986	3.020	2.948	3.018	2.942	3.012

ACLOC										
	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$
$\Phi_1$	1.618	1.619	1.618	1.618	1.618	1.619	1.618	1.619	1.617	1.619
$\Phi_2$	1.999	2.002	2.000	2.000	2.000	2.002	1.999	2.001	1.997	2.002
$\Phi_3$	2.414	2.418	2.416	2.417	2.416	2.422	2.411	2.428	2.408	2.417
$\Phi_4$	2.962	3.007	3.024	2.996	2.986	3.02	2.948	3.018	2.942	3.012

ECLOC										
	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$
$\Phi_1$	1.618	1.619	1.618	1.618	1.618	1.619	1.618	1.619	1.618	1.618
$\Phi_2$	1.999	2.002	2.000	2.000	2.000	2.004	1.999	2.000	1.998	2.002
$\Phi_3$	2.414	2.417	2.416	2.416	2.415	2.419	2.413	2.419	2.410	2.416
$\Phi_4$	2.977	3.004	3.014	2.998	2.992	3.012	2.968	3.011	2.965	3.009

PCLOC										
	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$
$\Phi_1$	1.619	1.618	1.618	1.618	1.618	1.618	1.619	1.618	1.618	1.618
$\Phi_2$	2.002	2.000	2.001	2.001	2.001	2.000	2.000	2.001	2.000	2.000
$\Phi_3$	2.417	2.415	2.416	2.417	2.419	2.416	2.417	2.418	2.415	2.416
$\Phi_4$	2.999	3.001	3.003	3.003	3.000	3.002	2.999	3.014	2.991	3.007

Taula 5.3: Resultats dels ordres locals de convergència ACLOC, CLOC, ECLOC i PCLOC per als mètodes iteratius  $\Phi_i$  amb  $1 \leq i \leq 4$  aplicats a les funcions  $f_j(x)$  amb  $1 \leq j \leq 10$ .

elevat d'iterats i el treball a realitzar en casa iteració no és prou per poder fer una mesura acurada dels temps sense que processos, externs als càlculs, del ordinador interfereixin.

	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$
$\Phi_1$	3.8821	1.7891	2.8078	2.5109	0.1736	4.3558	0.2608	0.1993	3.1027	3.2462
$\Phi_2$	6.1488	2.3348	4.2323	3.6006	0.1274	6.1947	0.2441	0.1640	3.5086	4.4025
$\Phi_3$	4.1346	1.8939	2.9834	2.7561	0.1198	4.4479	0.2104	0.1274	2.9844	3.4596
$\Phi_4$	5.8487	2.3427	4.7375	3.7361	0.1063	5.5802	0.2701	0.1577	3.8783	4.1499

Taula 5.4: Pendants de les rectes de regressió de la figura 5.2 per mètode i funció amb una aritmètica de 4096 dígit.

	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$
$\Phi_1$	90.9774	46.6537	80.9215	50.2588	2.5840	122.3549	4.5992	3.0127	48.6218	99.0855
$\Phi_2$	139.8170	63.8318	130.0034	70.8335	2.4120	175.0226	5.1866	3.1279	69.4779	138.6570
$\Phi_3$	97.7010	51.2645	93.6165	55.3009	2.2093	130.9511	4.4047	2.5570	52.6778	104.2369
$\Phi_4$	131.5765	63.3157	127.5082	72.7140	2.0845	164.8307	5.3793	2.8692	67.3910	128.8950

Taula 5.5: Pendants de les rectes de regressió de la figura 5.2 per mètode i funció amb una aritmètica de 32784 dígit.

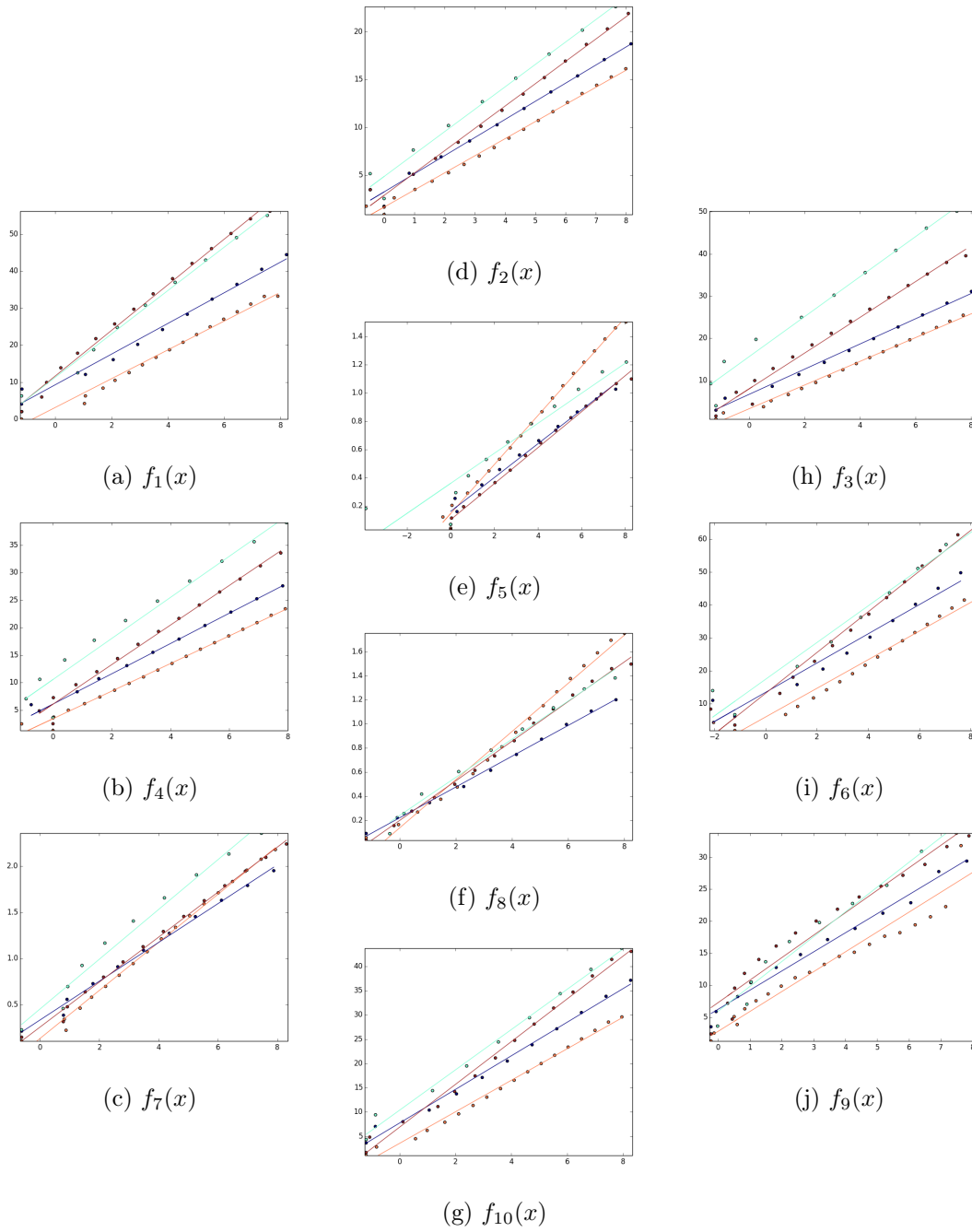


Figura 5.2: Rectes de regressió dels temps per iteració de les funcions  $f_j(x)$ ,  $1 \leq j \leq 10$ , per als esquemes iteratius  $\Phi_i$ ,  $1 \leq i \leq 4$ , amb una aritmètica de 4096 dígit. Les ordenades representen el temps acumulat en mil·lisegons de les iteracions i les abscisses  $\log \hat{D}_n$ .

## 5.2 Resultats amb aritmètica adaptativa

Les taules generades per les resolucions de les funcions test  $f_j(x)$  amb  $1 \leq j \leq 10$  amb els mètodes  $\Phi_i$  amb  $1 \leq i \leq 4$  amb aritmètica adaptativa es troben a l'annex A.2.

A la taula 5.6 es veu un resum de les resolucions de les funcions  $f_j(x)$  amb  $1 \leq j \leq 10$  amb els mètodes  $\Phi_i$  amb  $1 \leq i \leq 4$  amb el càlcul de les prediccions  $m_I$  (2.7.1),  $\hat{m}_I$  (2.7.2),  $\tilde{m}_I$  (2.7.3) i  $\check{m}_I$  (2.7.4).

Per a cada funció i mètode es presenta el número d'iteracions  $I$ , el número de dígitos correctes en la  $I$ -ésima iteració  $D_I$  i el número d'iteracions  $\psi = I - \max n$  t.q.  $\%t_n \geq 0.8$ , és a dir, el número d'iteracions que acumulen un 80% del temps total d'execució. EL valor  $\%t_n$  indica el temps acumulat de les iteracions  $x_n, \dots, x_I$  expressat en tan per 1.

La taula 5.7 mostra el número d'iteracions promig  $\bar{N}$ , el número mitjà d'iteracions amb un temps acumulat més gran a 0.8  $\bar{\psi}$  i el ratio entre tots dos.

Considerant que el cost de la resolució d'una funció amb aritmètica adaptativa es pot representar com el temps de les últimes  $\bar{\psi}$  iteracions. El ratio  $\bar{N}/\bar{\psi}$  dóna una aproximació de la diferència computacional entre aritmètica fixa i adaptativa per cadascun dels mètodes. Cal fer notar que el ratio  $\bar{N}/\bar{\psi}$  no serveix per comparar dos mètodes a causa de la diferència d'ordre de convergència entre ells.

El mètode  $\Phi_4$  no mostra tan bons resultats com la resta de mètodes. Aquest fet és degut a que l'aproximació inicial  $x_0$  escollida no es tan acurada com en la resta de mètodes ja que té un radi de convergència més estret.

Agafant les aproximacions inicials  $x_0$  com les 4 primeres xifres significatives de l'arrel  $\alpha$  s'aconsegueix un número d'iteracions promig  $\bar{N} = 6$  i un número mitjà d'iteracions amb un temps acumulat més gran a 0.8  $\bar{\psi} = 3$ . Les taules d'aquestes execucions es poden trobar a A.2.

Les prediccions  $m_I$  (2.7.1),  $\hat{m}_I$  (2.7.2),  $\tilde{m}_I$  (2.7.3) i  $\check{m}_I$  (2.7.4) són molt properes a  $D_I$  en la  $I$ -ésima iteració donant uns dígitos extres per a realitzar el càlcul de  $X_{I+1}$  amb precisió suficient.

$f_1(x), x_{-1} = 3.0, x_0 = 2.5$							
	$I$	$D_I$	$m_I$	$\hat{m}_I$	$\tilde{m}_I$	$\psi$	$\%t_n$
$\Phi_1$	16	2718	2723	2723	2722	2726	3 0.845
$\Phi_2$	13	2057	2065	2065	2063	2065	2 0.834
$\Phi_3$	9	3699	3710	3710	3706	3707	2 0.941
$\Phi_4$	7	1871	1900	1900	1902	1879	5 0.828

$f_3(x), x_{-1} = 0.5, x_0 = 1.0$							
	$I$	$D_I$	$m_I$	$\hat{m}_I$	$\tilde{m}_I$	$\psi$	$\%t_n$
$\Phi_1$	17	3654	3659	3659	3658	3663	3 0.877
$\Phi_2$	13	2480	2488	2488	2485	2488	2 0.877
$\Phi_3$	9	3040	3050	3050	3045	3049	1 0.806
$\Phi_4$	8	1797	1809	1809	1796	1807	5 0.812

$f_5(x), x_{-1} = 3.0, x_0 = 2.0$							
	$I$	$D_I$	$m_I$	$\hat{m}_I$	$\tilde{m}_I$	$\psi$	$\%t_n$
$\Phi_1$	17	3080	3085	3085	3083	3088	11 0.809
$\Phi_2$	12	3948	3956	3956	3953	3957	5 0.809
$\Phi_3$	9	1925	1936	1936	1931	1933	5 0.822
$\Phi_4$	9	3127	3151	3151	3150	3135	3 0.818

$f_7(x), x_{-1} = 1.0, x_0 = 1.29$							
	$I$	$D_I$	$m_I$	$\hat{m}_I$	$\tilde{m}_I$	$\psi$	$\%t_n$
$\Phi_1$	16	2786	2792	2792	2790	2794	10 0.831
$\Phi_2$	12	4045	4055	4055	4053	4053	4 0.813
$\Phi_3$	9	2638	2651	2651	2648	2645	3 0.807
$\Phi_4$	8	1741	1773	1773	1779	1749	6 0.842

$f_9(x), x_{-1} = 3.10, x_0 = 2.995$							
	$I$	$D_I$	$m_I$	$\hat{m}_I$	$\tilde{m}_I$	$\psi$	$\%t_n$
$\Phi_1$	19	3356	3362	3362	3361	3363	3 0.803
$\Phi_2$	14	2661	2672	2672	2670	2668	2 0.843
$\Phi_3$	10	2486	2501	2501	2500	2493	2 0.908
$\Phi_4$	8	1794	1828	1828	1835	1801	6 0.801

$f_2(x), x_{-1} = 3.0, x_0 = 2.5$							
	$I$	$D_I$	$m_I$	$\hat{m}_I$	$\tilde{m}_I$	$\psi$	$\%t_n$
$\Phi_1$	16	2942	2946	2946	2944	2951	3 0.842
$\Phi_2$	11	3200	3205	3205	3200	3207	2 0.906
$\Phi_3$	9	3490	3498	3498	3491	3499	2 0.920
$\Phi_4$	7	2101	2117	2117	2108	2111	5 0.841

$f_4(x), x_{-1} = 2.0, x_0 = 1.0$							
	$I$	$D_I$	$m_I$	$\hat{m}_I$	$\tilde{m}_I$	$\psi$	$\%t_n$
$\Phi_1$	17	2713	2718	2718	2716	2722	3 0.824
$\Phi_2$	12	2326	2334	2334	2331	2334	2 0.860
$\Phi_3$	9	2483	2494	2494	2488	2492	2 0.923
$\Phi_4$	9	2820	2839	2839	2834	2828	6 0.809

$f_6(x), x_{-1} = 1.5, x_0 = 1.0$							
	$I$	$D_I$	$m_I$	$\hat{m}_I$	$\tilde{m}_I$	$\psi$	$\%t_n$
$\Phi_1$	16	3751	3755	3755	3753	3760	2 0.800
$\Phi_2$	12	3612	3619	3619	3611	3623	2 0.919
$\Phi_3$	8	2017	2025	2025	2018	2026	2 0.936
$\Phi_4$	7	3424	3432	3432	3416	3435	3 0.833

$f_8(x), x_{-1} = 1.0, x_0 = 0.8$							
	$I$	$D_I$	$m_I$	$\hat{m}_I$	$\tilde{m}_I$	$\psi$	$\%t_n$
$\Phi_1$	15	2996	2998	2998	2996	3006	9 0.803
$\Phi_2$	12	3789	3796	3796	3793	3797	5 0.812
$\Phi_3$	8	2214	2219	2219	2209	2224	6 0.838
$\Phi_4$	7	2146	2159	2159	2146	2156	3 0.822

$f_{10}(x), x_{-1} = 2.5, x_0 = 2.0$							
	$I$	$D_I$	$m_I$	$\hat{m}_I$	$\tilde{m}_I$	$\psi$	$\%t_n$
$\Phi_1$	16	2862	2866	2866	2865	2871	3 0.861
$\Phi_2$	12	3981	3988	3988	3982	3991	2 0.905
$\Phi_3$	9	3920	3929	3929	3923	3929	1 0.819
$\Phi_4$	7	2838	2850	2850	2839	2847	3 0.805

Taula 5.6: Execucio de les 10 funcions mostra per als algorismes  $\Phi_i$  amb  $1 \leq i \leq 4$  amb aritmètica adaptativa.

MI	$\bar{N}$	$\bar{\psi}$	$\bar{N}/\bar{\psi}$
$\Phi_1$	16.0	3.0	5.3
$\Phi_2$	12.0	2.0	6.0
$\Phi_3$	9.0	2.0	4.5
$\Phi_4$	7.0	5.0	1.4

Taula 5.7: Ratio del cost computacional entre AA i AF per als mètodes  $\Phi_i$  amb  $1 \leq i \leq 4$  observat en la resolució de les funcions test  $f_j(x)$  amb  $1 \leq j \leq 10$ .



### 5.3 Comparativa entre AF i AA

Les taules comparatives entre AF i AA generades per les resolucions de les funcions test  $f_j(x)$  amb  $1 \leq j \leq 10$  amb els mètodes  $\Phi_i$  amb  $1 \leq i \leq 4$  es troben a l'annex A.3.

En aquesta secció es comparen diferents aspectes de la resolució de les funcions mostra  $f_j(x)$  amb  $1 \leq j \leq 10$  amb els mètodes  $\Phi_i$  amb  $1 \leq i \leq 4$  amb aritmètica fixa (AF) i aritmètica adaptativa (AA). Els aspectes que es comparen són:

- Temps mitjà d'execució i *speedup*.
- Error

Per la comparativa entre els temps d'execució entre aritmètica fixa i adaptativa es fan 1000 execucions per mètode, funció i tipus d'aritmètica. A la taula 5.8 es poden veure els resultats. Per tots els casos sense excepció la resolució mitjançant aritmètica adaptativa és més ràpida que la resolució mitjançant aritmètica fixa.

		$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$
$\Phi_1$	AF	37.566158	17.323770	29.813241	25.67815	1.609922	49.03753	2.163253	1.716848	27.054516	33.680320
	AA	6.3331840	3.5529650	6.746386	3.39539	0.499694	11.635561	0.592169	0.599773	3.195396	4.9479020
$\Phi_2$	AF	61.425420	22.691870	44.19821	35.60726	1.387392	72.16463	2.487315	1.888689	36.0382	48.974360
	AA	8.5360580	9.8762560	7.35044	5.620946	0.615261	27.991572	1.083664	0.825816	5.152087	21.679984
$\Phi_3$	AF	45.930850	18.936430	32.41175	28.87361	1.213107	52.15957	2.070572	1.452878	29.58668	40.653630
	AA	27.808007	9.2586880	13.926815	7.63135	0.377623	11.039683	0.798198	0.560978	6.361933	27.598178
$\Phi_4$	AF	52.093730	21.907230	52.29815	40.23989	1.423887	64.09881	2.393706	1.548816	31.65559	43.873570
	AA	28.641400	13.425415	24.934982	22.95564	0.723285	59.38833	1.127744	0.680268	14.100514	30.388830

Taula 5.8: Temps mitjà d'execució AA i AF (ms).

A les figures 5.3 i 5.4 es veuen gràficament els resultats obtinguts. D'una banda a la figura 5.3 es veuen els temps d'execució per mètode i funció i el *speedup* aconseguit mitjançant la resolució amb aritmètica adaptativa per cada mètode i funció. D'altra banda a la figura 5.4 s'observen dos *boxplots*. El primer mostra els resultats de *speedup* per mètode, el segon mostra els resultats de *speedup* per funció.

En general amb AA aconseguim un factor mitjà de millora de 3.5 respecte AF, però degut a la variabilitat per funció i mètode que s'observa a la figura 5.4 i la presència d'alguns *outliers* en les funcions  $f_2(x)$  i  $f_{10}(x)$  el factor de millora s'hauria de considerar per mètode o per funció tal i com es mostra en els *boxplots*.

Per la comparativa dels errors entre aritmètica fixa i adaptativa per cada mètode i funció es calcula  $|e_{IAF} - e_{IAA}|$  per veure la diferència entre el nombre de dígitos correctes. Els resultats es veuen a la taula 5.9.

	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$
$\Phi_1$	3.79e-2731	7.94e-2959	1.14e-3669	6.24e-2725	1.80e-3091	4.53e-3348	7.16e-4035	6.46e-2593	1.47e-3369	1.43e-2875
$\Phi_2$	4.65e-2074	2.20e-3219	4.31e-2502	4.66e-2343	1.15e-3962	2.41e-3123	7.94e-3819	5.45e-2987	5.29e-2678	3.97e-3995
$\Phi_3$	4.67e-3719	9.98e-3514	4.40e-3065	8.49e-2504	6.09e-1946	8.13e-1873	4.74e-2094	2.45e-2151	3.92e-2510	7.07e-3940
$\Phi_4$	2.36e-1897	8.25e-2129	5.60e-1821	9.29e-3347	2.29e-1902	1.82e-3451	3.13e-2422	2.45e-2198	1.99e-2787	3.75e-2864

Taula 5.9:  $|e_{I_{AF}} - e_{I_{AA}}|$ .

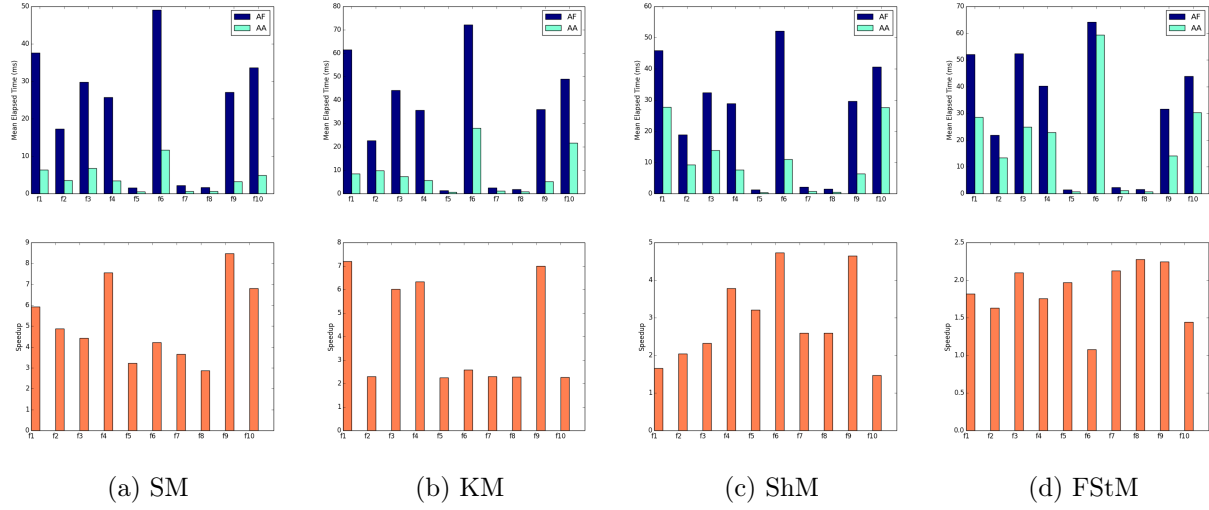


Figura 5.3: Temps d'execució en ms i *speedup* per les funcions  $f_j(x)$ ,  $1 \leq j \leq 10$ , i mètodes  $\Phi_i$ ,  $1 \leq i \leq 4$ .

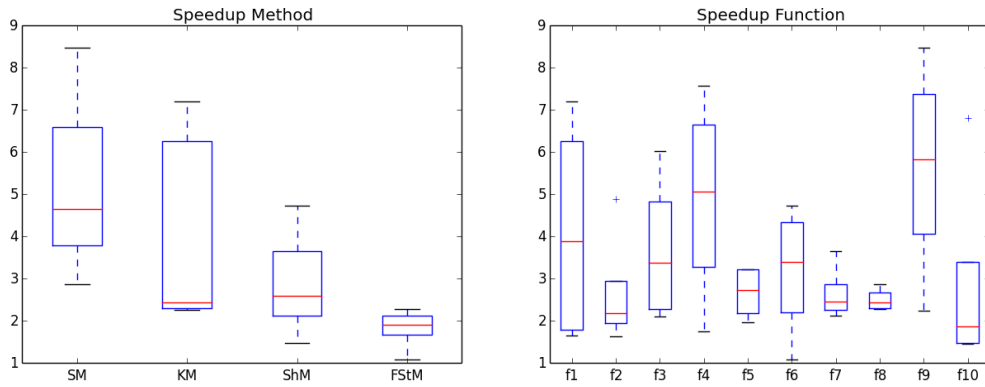


Figura 5.4: Boxplot de l'*speedup* de MPFR respecte BigFloat (esquerra) i ARPREC (dreta).

## 5.4 Aproximacions inicials i Número d'iteracions

Es vol veure des d'un punt de vista pràctic la relació entre el número d'iteracions, el número de decimals correctes i les aproximacions inicials.

Per aquest fi es resolen les equacions  $f_j(x) = 0$  amb  $1 \leq j \leq 10$  amb els mètodes iteratius  $\Phi_i$  amb  $1 \leq i \leq 4$ , a partir de les aproximacions inicials  $x_{-1} = \alpha - 0.01$  i  $x_0 = \alpha + 0.01$  pels mètodes  $\Phi_i$ ,  $1 \leq i \leq 3$ , i  $x_0 = \alpha + 0.01$  per  $\Phi_4$ . Les aproximacions inicials per la  $k$ -resolució es calculen de la següent manera  $x_{-1} = \alpha - k * 0.01$  i  $x_0 = \alpha + k * 0.01$ . En el cas particular de  $\Phi_4$  es faran  $k$  resolucions a partir de l'aproximació inicial  $x_0 = \alpha + k * 0.01$ . S'anirà augmentant el valor de  $k$  mentre el nombre d'iteracions sigui menor a 20 i el mètode convergeixi. Aquesta és una de les moltes possibles variacions de les aproximacions inicials que es poden realitzar. A l'estudi teòric realitzat a la secció 5.5 es contemplen dues més.

Els resultats d'aquests càlculs es veuen a les figures 5.5-5.14. Per a cada funció i mètode es veu en una mateixa tonalitat de color el conjunt d'aproximacions inicials amb les quals el mètode resol l'equació no lineal amb el mateix número d'iteracions. La línia en color negre indica per cada aproximació inicial el nombre de decimals correctes amb el què el mètode convergeix. L'ordenada  $d_n$  de la gràfica en negre és el número de decimals correctes aconseguits. El número màxim i mínim de decimals està marcat amb un punt.

Es pot veure a les figures 5.5-5.14 que, en general, dins de cada tonalitat de color, que representa un número  $n$  d'iteracions, el número de decimals correctes aconseguits pel mètode de resolució va decreixent a mesura que s'apropa al nou interval, on torna a augmentar.

A causa de les característiques particulars de cada funció els resultats són molt variats i no tots els mètodes es comporten de la mateixa manera per a una determinada funció. Alguns exemples d'aquestes particularitats són la diferència de la longitud dels intervals de convergència per a cada mètode i funció i les aproximacions inicials que ens porten a trobar el nombre màxim de xifres significatives.

A la taula 5.10 es veuen per cada funció i mètode les aproximacions inicials amb les quals el mètode convergeix amb el mínim número d'iterats i el màxim número de dígitos correctes. Cal destacar que aquesta condició es troba per totes les funcions, menys per la funció  $f_7(x)$ , a l'interval més proper, és a dir, per les aproximacions inicials  $x_{-1} = \alpha - 0.01$  i  $x_0 = \alpha + 0.01$  pels mètodes  $\Phi_i$ ,  $1 \leq i \leq 3$ , i  $x_0 = \alpha + 0.01$  per  $\Phi_4$ . La funció  $f_7(x)$  té algunes particularitats que fa que els mètodes  $\Phi_i$ ,  $2 \leq i \leq 4$  tinguin una aproximació inicial amb el número mínim d'iterats i màxim número de dígitos d'una grandària més elevada que per la resta de funcions.

A la taula 5.11 es veuen per a cada funció i mètode les aproximacions inicials amb les quals el mètode convergeix amb el màxim número de xifres significatives. Sota aquesta condició cada funció i mètode presenta unes condicions de longitud de l'interval i un número de dígitos correctes molt variats.

$f_i(x)$	$\Phi_1$				$\Phi_2$				$\Phi_3$				$\Phi_4$		
	n	$D_I$	$x_{-1}$	$x_0$	n	$D_I$	$x_{-1}$	$x_0$	n	$D_I$	$x_{-1}$	$x_0$	n	$D_I$	$x_0$
$f_1(x)$	16	3436	2.49	2.51	11	2335	2.49	2.51	9	2848	2.49	2.51	8	2369	2.51
$f_2(x)$	15	2764	1.75	1.77	11	3153	1.75	1.77	9	3893	1.75	1.77	7	1774	1.77
$f_3(x)$	16	3834	0.61	0.63	11	2408	0.61	0.63	9	3305	0.61	0.63	7	1440	0.63
$f_4(x)$	16	3658	1.65	1.67	11	2257	1.65	1.67	9	3397	1.65	1.67	8	2986	1.67
$f_5(x)$	16	3589	2.88	2.90	11	2369	2.88	2.90	9	3120	2.88	2.90	8	2975	2.90
$f_6(x)$	15	2961	1.73	1.75	11	3075	1.73	1.75	9	4033	1.73	1.75	7	2677	1.75
$f_7(x)$	16	3675	1.28	1.30	11	3999	-8.72	11.3	7	3093	-8.72	11.3	7	3584	3.47
						3138	1.28	1.30	9	2742	1.28	1.30	8	3531	1.30
$f_8(x)$	15	3625	0.86	0.88	11	3320	0.86	0.88	8	2052	0.86	0.88	7	2208	0.88
$f_9(x)$	17	3060	2.99	3.01	12	2194	2.99	3.01	9	1876	2.99	3.01	10	2641	3.01
$f_{10}(x)$	16	4074	2.37	2.39	11	2677	2.37	2.39	9	4003	2.37	2.39	7	4077	2.39

Taula 5.10: Intervals de convergència amb el número mínim d'iterats i màxim número de dígitos correctes per les funcions  $f_i(x)$ ,  $1 \leq i \leq 10$ , per els mètodes  $\Phi_j$ ,  $1 \leq j \leq 4$ .

$f_i(x)$	$\Phi_1$				$\Phi_2$				$\Phi_3$				$\Phi_4$		
	n	$D_I$	$x_{-1}$	$x_0$	n	$D_I$	$x_{-1}$	$x_0$	n	$D_I$	$x_{-1}$	$x_0$	n	$D_I$	$x_0$
$f_1(x)$	19	4081	2.26	2.74	17	4083	1.56	3.44	12	4080	1.88	3.12	10	4070	2.54
$f_2(x)$	19	4080	0.44	3.08	15	4095	-24.81	28.33	10	4091	1.08	2.44	10	4091	6.33
$f_3(x)$	19	4036	0.16	1.08	13	4007	0.50	0.74	10	4079	0.45	0.79	9	4033	0.74
$f_4(x)$	19	4086	1.17	2.15	18	4091	-1.09	4.41	11	4087	0.98	2.34	10	4089	2.46
$f_5(x)$	19	4052	2.49	3.29	18	4096	-3.11	8.89	11	4034	2.33	3.45	13	4060	3.40
$f_6(x)$	17	4087	1.57	1.91	14	4064	1.07	2.41	11	4083	0.00	3.48	8	4069	1.99
$f_7(x)$	19	3991	1.19	1.39	14	4090	0.71	1.87	8	4081	-1.15	3.73	8	4094	5.31
$f_8(x)$	17	4069	0.57	1.17	12	4068	0.82	0.92	19	4069	-0.66	2.40	16	4089	1.26
$f_9(x)$	19	3985	2.96	3.04	17	4024	2.84	3.16	13	3852	2.87	3.13	13	2950	3.02
$f_{10}(x)$	18	4094	1.99	2.77	16	4071	1.09	3.67	12	4085	0.31	4.45	9	4077	6.30

Taula 5.11: Intervals de convergència amb el màxim número de dígitos correctes per les funcions  $f_i(x)$ ,  $1 \leq i \leq 10$ , per els mètodes  $\Phi_j$ ,  $1 \leq j \leq 4$ .

$f_i(x)$	$\Phi_1$					$\Phi_2$					$\Phi_3$					$\Phi_4$			
	n	$D_I$	$x_{-1}$	$x_0$	$\Delta$	n	$D_I$	$x_{-1}$	$x_0$	$\Delta$	n	$D_I$	$x_{-1}$	$x_0$	$\Delta$	n	$D_I$	$x_0$	$\Delta$
$f_1(x)$	20	4007	2.08	2.92	0.84	20	4035	0.53	4.47	3.94	15	3409	1.62	3.38	1.76	23	1595	2.63	0.13
$f_2(x)$	20	4070	-0.02	3.54	3.56	21	3915	-32.64	36.16	86.8	13	2374	-0.65	4.17	4.82	10	2542	11.77	10.01
$f_3(x)$	20	3996	-0.06	1.30	1.36	21	2420	0.20	1.04	0.84	21	1976	-0.89	2.13	3.02	20	1870	0.93	0.31
$f_4(x)$	20	4050	0.78	2.54	1.76	20	4086	-2.77	6.09	8.86	16	2349	-0.02	3.34	3.36	10	3482	11.67	10.01
$f_5(x)$	20	4043	2.21	3.57	1.36	20	4093	-11.34	17.12	28.46	27	2038	1.67	4.11	2.44	20	3528	4.01	1.12
$f_6(x)$	20	3851	-0.06	3.54	3.60	20	3599	0.40	3.08	2.68	22	3220	-0.84	4.32	5.16	10	2867	3.25	1.51
$f_7(x)$	20	3902	1.12	1.46	0.34	11	3999	-8.72	11.30	20.02	7	3093	-8.72	11.3	20.02	7	2351	11.30	10.01
$f_8(x)$	20	3141	0.23	1.51	1.28	21	3922	0.57	1.17	0.60	20	4029	-0.84	2.58	3.42	20	2519	1.36	0.49
$f_9(x)$	20	3869	2.93	3.07	0.14	20	3704	2.65	3.35	0.70	14	2841	2.86	3.14	0.28	23	1842	3.03	0.03
$f_{10}(x)$	20	4093	0.84	3.92	3.08	31	3432	0.94	3.82	2.88	26	2359	0.08	4.68	4.60	20	2485	11.17	8.79

Taula 5.12: Intervals de convergència de radi més gran per les funcions  $f_i(x)$ ,  $1 \leq i \leq 10$ , per els mètodes  $\Phi_j$ ,  $1 \leq j \leq 4$ .

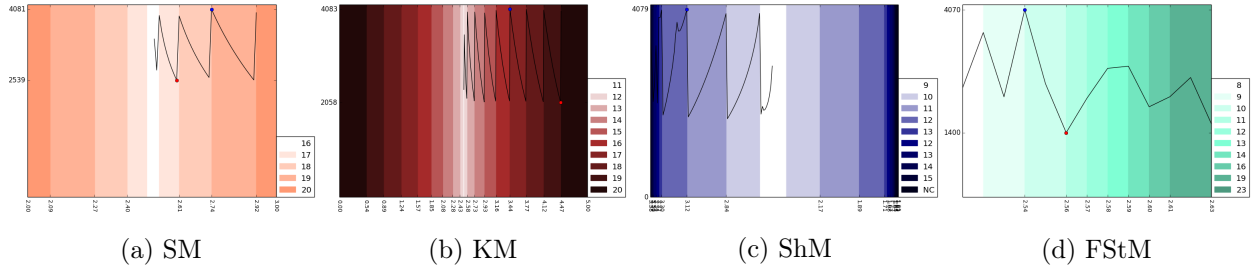


Figura 5.5:  $f_1(x)$

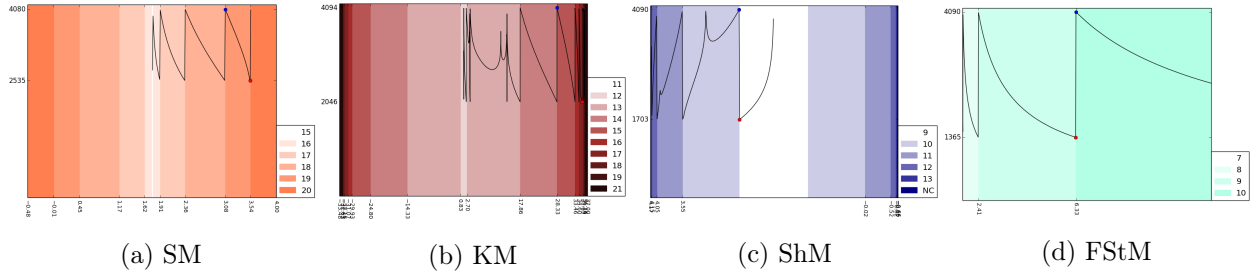


Figura 5.6:  $f_2(x)$

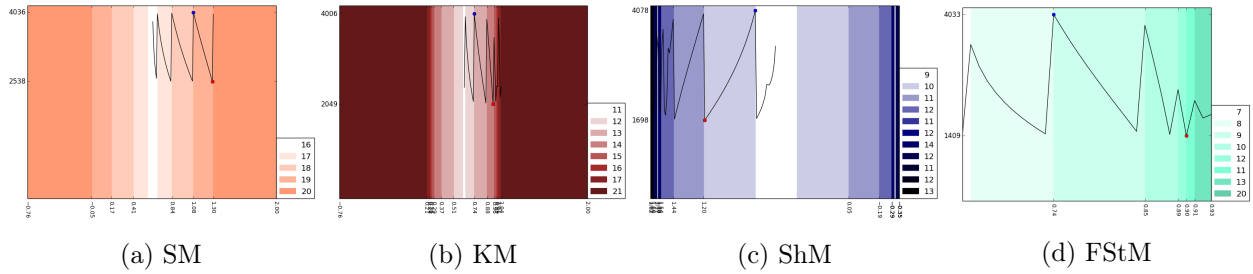


Figura 5.7:  $f_3(x)$

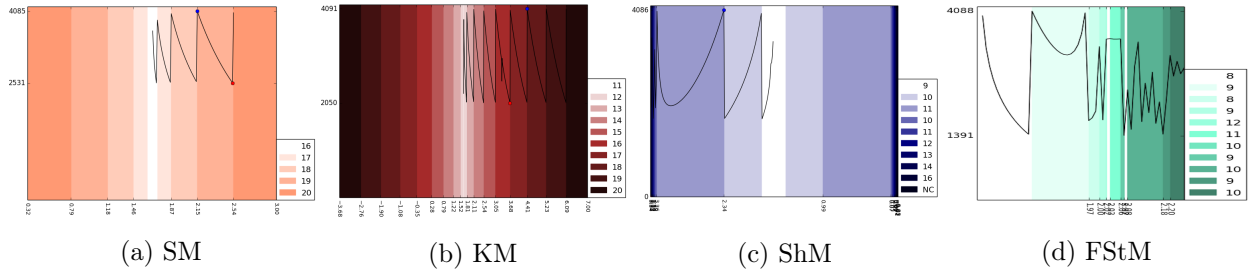


Figura 5.8:  $f_4(x)$

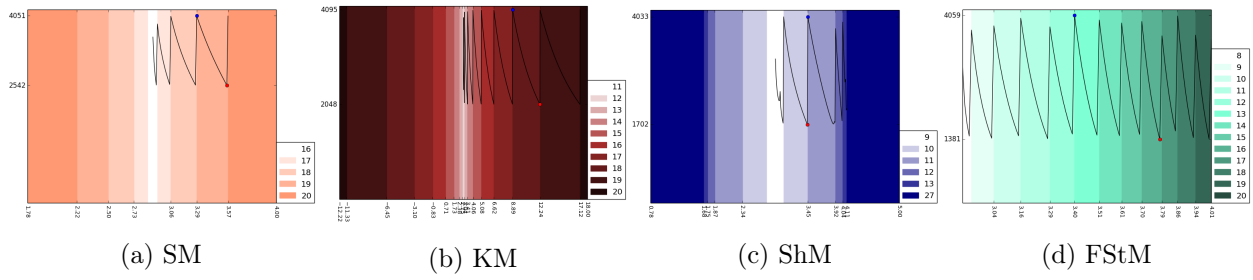


Figura 5.9:  $f_5(x)$

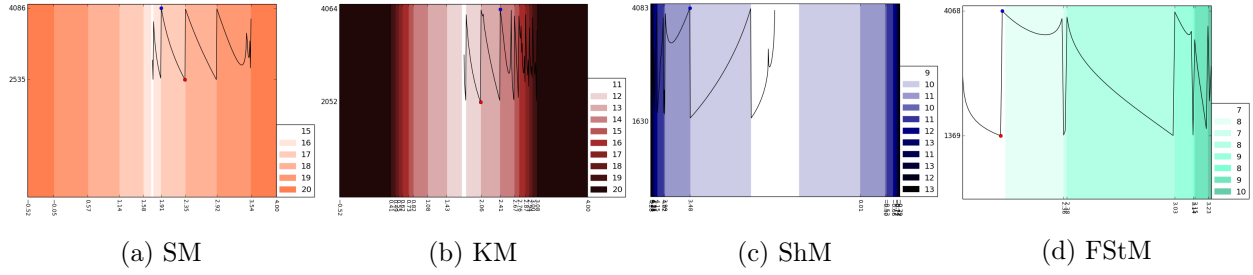


Figura 5.10:  $f_6(x)$

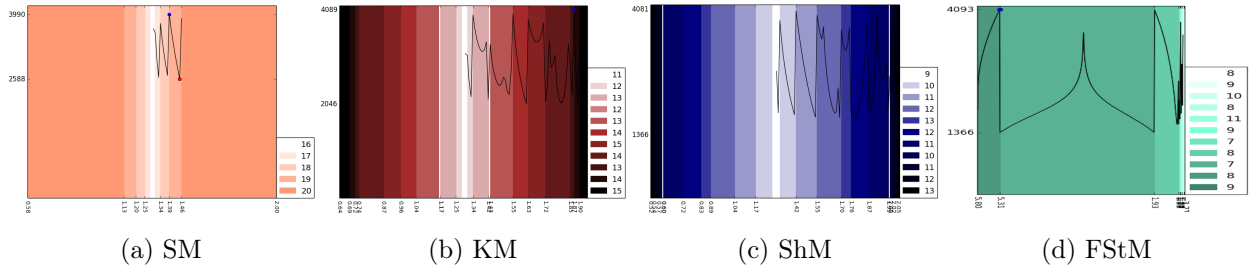


Figura 5.11:  $f_7(x)$

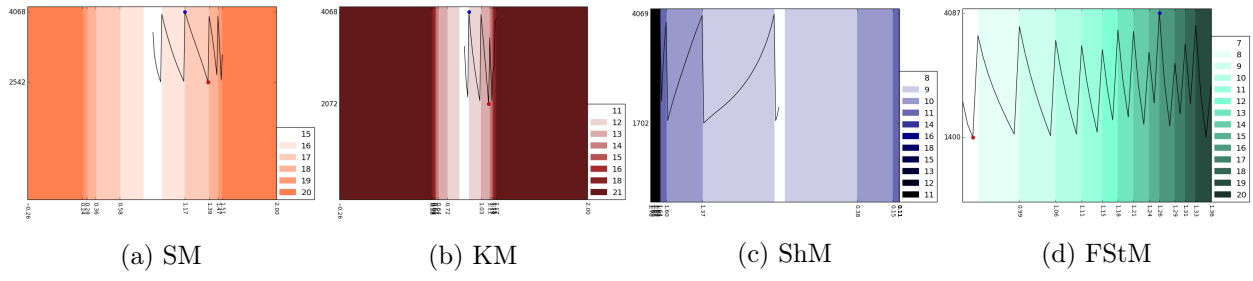


Figura 5.12:  $f_8(x)$

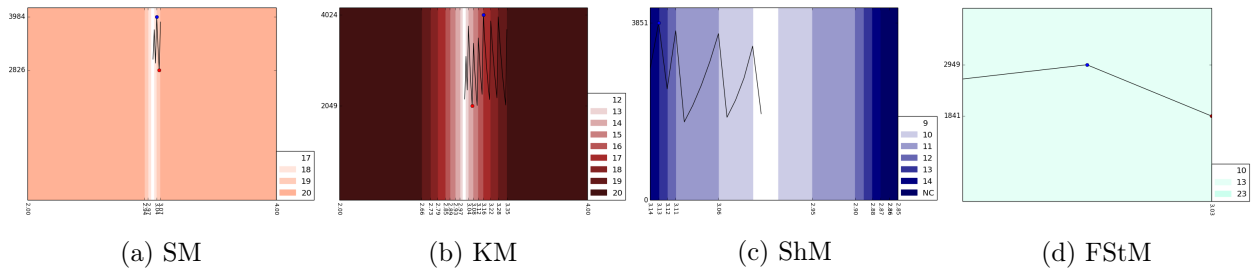


Figura 5.13:  $f_9(x)$

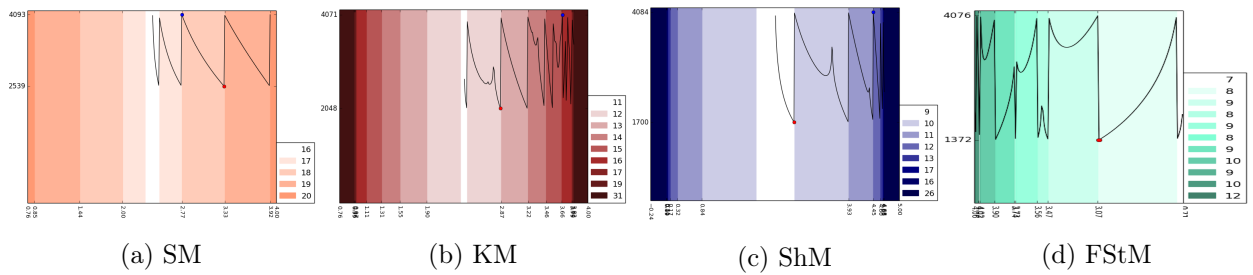


Figura 5.14:  $f_{10}(x)$

## 5.5 Càlcul del radi de convergència a partir de K, L i M

Des d'un punt de vista teòric es proposa un estudi de la grandària del radi de convergència pels mètodes  $\Phi_j$ , per  $1 \leq j \leq 4$ , per a les 10 funcions mostra presentades a la taula 5.1 (secció 5).

Per a cada funció i cas d'estudi es calculen  $K$  (4.2.1),  $L$  (4.2.18) i  $M$  (4.2.21) de la manera següent:

$$K = \frac{|f'(\alpha)^{-1}([x, y] - [u, v])|}{(|x - u| + |y - v|)}, \quad (5.5.1)$$

$$L = \frac{|f'(\alpha)^{-1}([x_{n-1}, 2x_n - x_{n-1}] - [x_n, x_n])|}{|x_n - x_{n-1}|^2}, \quad (5.5.2)$$

$$M = \max_{x \in J} |f'(x)|, \quad (5.5.3)$$

on  $J$  és l'interval més gran amb un nombre d'iteracions menor a 20, criteri que s'ha establert empíricament per determinar que el mètode convergeix amb un nombre correcte d'iteracions.

L'objectiu és poder aproximar la grandària del radi de convergència,  $r_{\Phi_i}$ , de cada esquema iteratiu,  $\Phi_i$ , a partir de 4.2.10. Com a cas especial s'ha afegit l'estudi del radi del mètode de Traub,  $\Phi_{3bis}$ , amb ordre de convergència  $1 + \sqrt{2}$ .

Es pendrà  $K$  com el valor màxim de 5.5.1 en variar  $x_0$ ,  $x_1$ ,  $u$  i  $v$  de les tres possibles següents formes:

$$(A) \quad x_0 \leq u \leq \alpha \leq v \leq x_1,$$

$$(B) \quad x_0 \leq u \leq v \leq x_1 \leq \alpha,$$

$$(C) \quad \alpha \leq x_0 \leq u \leq v \leq x_1,$$

On  $x_0$ ,  $x_1$  són les aproximacions inicials, extrems de l'interval amb màxim número d'iteracions i màxim número de dígitos correctes  $J$  trobat a la secció 5.4,  $\alpha$  l'arrel de la funció i  $\forall u, v \in B_r(\alpha)$ .

Per al cas (A) fixats  $x_0$  i  $x_1$  en els extrems de l'interval  $J$ , es pren un  $u$  entre  $x_0$  i  $\alpha$  i, llavors, es fa variar  $v$  entre  $\alpha$  i  $x_1$ .

$$v = x_1 - \frac{j}{10}(x_1 - \alpha), \quad 1 \leq j \leq 9,$$

El valor de  $u$  també es fa variar

$$u = x_0 + \frac{i}{10}(\alpha - x_0), \quad 1 \leq i \leq 9,$$

un cop s'han calculat els valors de  $K$  per l'anterior  $u$ .

Per al cas (B) fixades les aproximacions inicials  $x_0$  i  $x_1$ , que s'obtenen de la següent manera:  $x_0 = \alpha - 0.02$  i  $x_1 = \alpha - 0.01$ , es fa variar  $u$  i  $v$  entre  $x_0$  i  $x_1$  mentre  $u < v$ .



Per al cas (C) fixades les aproximacions inicials  $x_0$  i  $x_1$  calculades com  $x_0 = \alpha + 0.01$  i  $x_1 = \alpha + 0.02$ , es fan variar  $u$  i  $v$  com en el cas (B).

Es pendrà  $M$  (5.5.3) com el valor màxim de la derivada de la funció en l'interval  $J$ .

Es prendrà  $L$  com el valor màxim de (5.5.2) en variar  $x_0$  i  $x_1$  de les tres formes proposades:

(A)  $x_0 < \alpha < x_1$ ,

(B)  $x_0 < x_1 < \alpha$ ,

(C)  $\alpha < x_0 < x_1$ .

Com en el cas (A) de l'estudi de  $K$ ,  $x_0$  i  $x_1$  són les aproximacions inicials, els extrems de l'interval  $J$ . Pel cas (A) de l'estudi de  $L$   $x_0$  i  $x_1$  es fan variar mentre  $x_{0_i} < \alpha < x_{1_j}$  de la forma

$$x_{0_i} = x_0 + \frac{i}{10}(\alpha - x_0), \quad 1 \leq i \leq 9,$$

$$x_{1_j} = x_1 - \frac{j}{10}(x_1 - \alpha), \quad 1 \leq j \leq 9,$$

Per als casos (B) i (C) es varia  $x_0$  i  $x_1$  seguint el mateix esquema que en el cas (A). Les aproximacions inicials són calculades de la mateixa manera que en els casos (B) i (C) de l'estudi de  $K$ .

Els resultats obtinguts es presenten a la taula 5.13.

$f_i(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$
$r_{\Phi_1}$	0.033441	0.937007	0.050598	6.1964e-05	0.198544	1.766962	14.390931	0.260789	0.005444	4.968735
$r_{\Phi_2}$	0.004399	0.434622	0.019145	7.3918e-05	0.213479	0.703164	0.367832	0.143358	0.001114	0.846364
$r_{\Phi_3}$	0.020065	0.562204	0.030359	3.7178e-05	0.119126	1.060177	8.634559	0.156473	0.003266	2.981241
$r_{\Phi_{3bis}}$	0.033441	0.937007	0.050598	6.1964e-05	0.198544	1.766962	14.390931	0.260789	0.005444	4.968735
$r_{\Phi_4}$	9.6163e-06	0.345097	0.012257	4.6473e-14	0.000382	0.638829	0.004265	0.000665	4.0240e-07	1.656245

Taula 5.13: Radis de convergència per a les funcions  $f_i(x)$ , per a  $1 \leq i \leq 10$  i els mètodes iteratius  $\Phi_j$  per a  $1 \leq j \leq 4$  fent servir les fórmules presentades a 4.2.10.

Tot i posar com a criteri de convergència un màxim de 20 iteracions, alguns mètodes i funcions arribarien a solució amb més iteracions en un interval de convergència amb radi major. De la taula 5.13 s'extreu la informació de que el mètode  $\Phi_j$  (FSTM) és el que té, en general, un interval més estret.

## 5.6 Comparativa entre MPFR, ARPREC i BigFloat

S'han realitzat tres implementacions amb els programaris MPFR [12], ARPREC [6] i BigFloat [7] del sistema de resolució d'equacions no lineals (Es pot veure el disseny del sistema de resolució a la secció 6). Aquests programes fan servir els algorismes  $\Phi_j$ ,  $1 \leq j \leq 4$ , per a la resolució de les funcions test  $f_i(x)$ ,  $1 \leq i \leq 10$ , presentades a la taula 5.1.

Es vol fer la comparativa dels tres programaris amb la mesura dels temps mitjans d'execució per a cada mètode i funció. Per mesurar els temps s'han realitzat 1000 execucions per mètode i funció per a cada programari i s'ha mesurat el temps mitjà d'execució. En les taules 5.14-5.16 es veuen els resultats obtinguts.

	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$
$\Phi_1$	31.4397	14.6823	24.4585	21.3447	1.06528	39.9062	1.72809	1.16127	30.9152	27.3167
$\Phi_2$	52.0509	20.4223	36.6205	30.2400	1.05912	60.0700	2.09129	1.46423	30.8464	40.6512
$\Phi_3$	39.2646	16.8738	27.6209	24.7870	0.97695	44.0650	1.78931	1.16221	27.4965	33.3050
$\Phi_4$	48.3841	19.9893	45.5705	35.6203	1.18345	56.9861	2.18328	1.33475	30.4584	38.6955

Taula 5.14: Temps mitjà d'execució (ms) en MPFR.

	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$
$\Phi_1$	1619.7217	360.9654	2829.6377	1437.6641	11.5844	4098.1745	17.9033	14.9376	666.3152	2806.4212
$\Phi_2$	2349.0595	537.5744	4061.7758	2084.9370	13.0571	6126.4953	26.1397	21.0919	905.8807	4244.8966
$\Phi_3$	2117.4784	445.3227	3721.7584	1670.9987	11.4594	5367.5423	19.1019	16.0468	798.3619	3693.4074
$\Phi_4$	2694.3978	599.8941	5281.0178	2372.0269	13.3383	6137.5075	25.6376	19.0153	897.2985	4216.1253

Taula 5.15: Temps mitjà d'execució (ms) en ARPREC.

	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$	$f_9(x)$	$f_{10}(x)$
$\Phi_1$	0.0593	20.5467	30.9767	28.2019	0.0468	42.1268	0.0422	0.0785	32.7108	33.7396
$\Phi_2$	0.0456	29.0657	47.9145	41.0213	0.0390	64.7862	0.0349	0.0603	45.8679	50.7380
$\Phi_3$	0.2921	24.4576	66.6253	41.9370	0.2238	52.8780	0.2079	0.0390	31.4202	32.4060
$\Phi_4$	0.0516	23.3333	50.2843	41.0125	0.0408	52.6772	0.0368	0.0684	36.1857	40.7108

Taula 5.16: Temps mitjà d'execució (ms) en BigFloat.

En les figures 5.15-5.16 es poden veure els resultats de les comparatives de temps d'execució mitjà i *speedup* gràficament.

En la figura 5.15 s'observa per a cada mètode  $\Phi_i$ ,  $1 \leq i \leq 4$ , i funció  $f_j(x)$ ,  $1 \leq j \leq 10$ , els temps d'execució mitjà (*Mean Ellapsed Time*) en ms i el *speedup* mesurat com el ratio  $\frac{time_{MPFR}}{time_t}$  on  $t = [\text{BigFloat}, \text{ARPREC}]$ .

En general MPFR és la llibreria més ràpida seguida de BigFloat i ARPREC. Les funcions  $f_1(x)$ ,  $f_5(x)$ ,  $f_7(x)$  i  $f_8(x)$  tenen un molt bon compartament en BigFloat; això és a causa de les seves característiques i algunes optimitzacions que realitza la llibreria.

En la figura 5.16 es veuen dos *boxplots* dels resultats de *speedup* de MPFR respecte BigFloat i ARPREC per als mètodes  $\Phi_i$ ,  $1 \leq i \leq 4$ . En aquest cas el *speedup* ha estat calculat com  $\frac{time_t}{time_{MPFR}}$  on  $t = [\text{BigFloat}, \text{ARPREC}]$  per tal de veure en quin factor és més ràpid. En aquesta figura s'observa que MPFR és més ràpid que BigFloat i ARPREC a causa que les dues medianes estan per sobre de 1.0. En el cas de BigFloat, MPFR és de mitja 1.06 vegades més ràpid. En el cas d'ARPREC i MPFR el factor és de 37.57. MPFR, per a aquestes funcions mostra i els mètodes de la Secant, Kurxatov, Shakno i Frozen Steffensen pot ser més ràpid en un factor entre 10.36 i 134.74 en el cas d'ARPREC i en un factor que varia entre 0.0008 i 2.41 en el cas de BigFloat.

BigFloat és un modul de Python que actua de *wrapper* de la llibreria MPFR <sup>1</sup> de C, és a dir, permet fer crides a la llibreria de C des de Python. És per aquest motiu que els temps d'execució entre MPFR i BigFloat siguin similars tot i que Python és un llenguatge interpretat i per tant més lent que C++. En el cas de BigFloat només s'està afegint un petit *overhead* del *wrapper* de Python. Per altres comparatives entre MPFR i d'altres eines de multiprecisió es pot trobar més informació a <sup>2</sup>.

---

<sup>1</sup>GNU MPFR library for arbitrary-precision floating-point reliable arithmetic.

<sup>2</sup><http://www.mpfr.org/mpfr-2.2.0/timings.html>

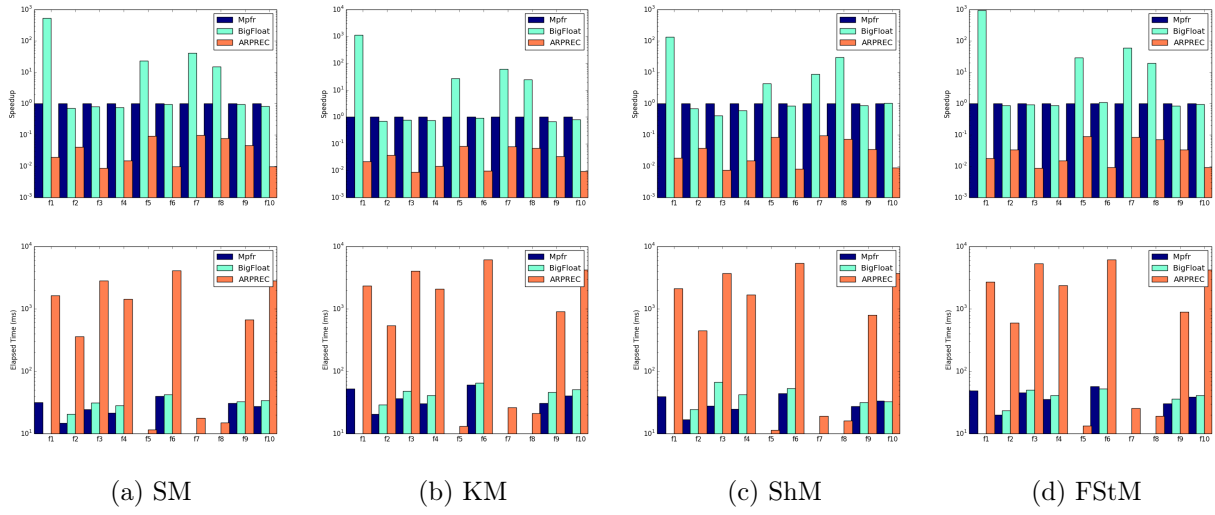


Figura 5.15: Temps d'execució en ms i *speedup* per les funcions  $f_j(x)$ ,  $1 \leq j \leq 10$ , i mètodes  $\Phi_i$ ,  $1 \leq i \leq 4$ .

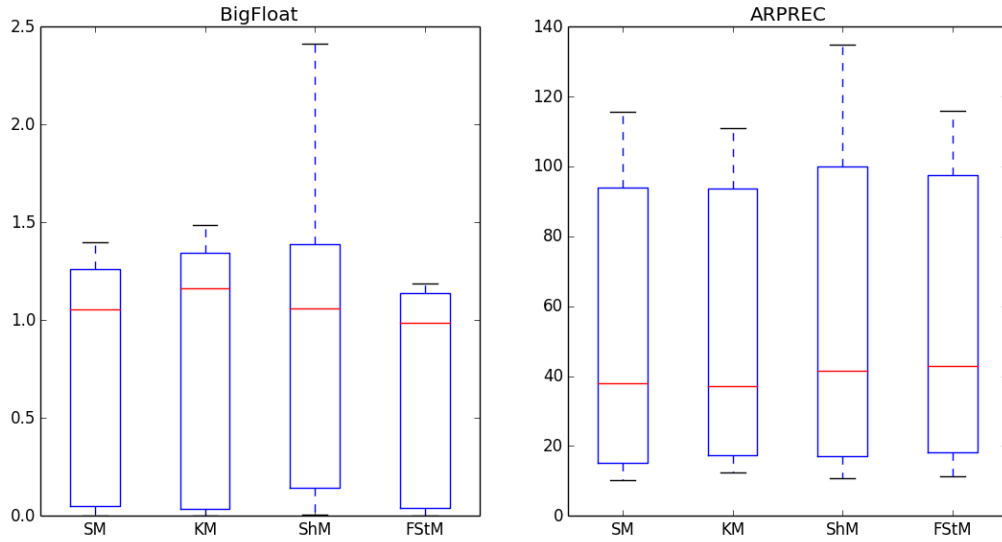


Figura 5.16: Boxplot de el *speedup* de MPFR respecte BigFloat i ARPREC.

## 5.7 Conclusions dels resultats numèrics

Amb aritmètica fixa les aproximacions són tan acurades que el número d'iterats  $I$  és el mateix per a tots quatre criteris de tall. Els decimals correctes  $D_I, \hat{D}_I, \tilde{D}_I, \check{D}_I$  són molt pròxims. Amb un error màxim de 3 dígits respecte  $D_I$ .

Els ordres de convergència local CLOC, ACLOC, ECLOC i PCLOC donen resultats amb 3 xifres significatives.

L'aritmètica adaptativa respecte l'aritmètica fixa comporta una millora considerable en els temps total de resolució. Les prediccions respecte la precisió necessària en la següent iteració són sempre molt properes al número de xifres significatives.

La diferència entre l'error  $I$ -èssim d'aritmètica adaptativa i fixa,  $|e_{I_{AF}} - e_{I_{AA}}|$ , és sempre més gran que la precisió aconseguida. Per tant, es pot dir que s'arriba a la  $I$ -èssima iteració amb la mateixa precisió en els resultats.

L'estudi pràctic sobre les aproximacions inicials mostra el pes que tenen les característiques particulars de cada funció i mètode en el número d'iteracions i la localitat de les aproximacions que proporcionen un número màxim i mínim de dígits correctes en la  $I$ -èssima iteració.

L'estudi dels radis de convergència fa present un altre criteri a tenir en compte respecte els mètodes iteratius i es que sota unes mateixes aproximacions inicials la convergència de tots els mètodes estudiats no es pot assegurar. Els mètode de Steffensen Congelat, en particular, té un radi de convergència molt estret.

De la comparativa entre les llibreries MPFR, BigFloat i ARPREC es pot concloure que MPFR és la més ràpida seguida de BigFloat. ARPREC és molt més lenta. Entre BigFloat i MPFR, BigFloat és més *friendly* i la diferència en termes de temps d'execució és menyspreable.

El fet de voler trobar un interval  $x_{-1}, x_0$  de convergència comú per tots els mètodes ha afectat en alguns resultats. En la aritmètica adaptativa, per exemple, l'aproximació inicial  $x_0$ , extrem del interval genèric  $x_{-1}, x_0$ , per al mètode d'Steffensen no és tan acurada com per als altres mètodes i comporta un nombre d'iterats superior al que es tindria amb una millor aproximació inicial.

Les mesures dels temps d'execució per aritmètica adaptativa i per les comparatives de temps entre aritmètica fixa i aritmètica adaptativa s'ha hagut de realitzar en un sistema operatiu *Linux* amb la distribució *Mint*. A causa que les execucions amb el sistema operatiu *OS X Yosemite (10.10.3)*

no son vàlides per les comparatives de temps d'execució. El sistemes *Turbo Boost*<sup>3</sup> i *Speed Step*<sup>4</sup> interfereixen en les execucions i no es poden desactivar. Amb la distribució *Linux Mint* no s'ha trobat el problema.

---

<sup>3</sup>Intel's propriety technology to run processors above their base frequency

<sup>4</sup><http://en.wikipedia.org/wiki/SpeedStep> per més detall

## Capítol 6

# Implementació

En la implementació principal del sistema de resolució d'equacions no lineals es fa servir la llibreria MPFR [12] de C++. Es fan dues implementacions més per a la llibreria ARPREC [6] de C++ i BigFloat [7] de Python per fer proves de *speedup*.

El codi font de totes tres implementacions es troba als annexos B.1.1, B.1.2 i B.1.3 respectivament. Per totes tres implementacions es farà servir el model orientat a objectes.

### 6.1 Disseny

El disseny de classes del sistema s'ha fet per permetre la màxima flexibilitat. L'objectiu és poder resoldre qualsevol funció amb qualsevol mètode o combinació de mètodes i avaluar l'error amb qualsevol paràmetre d'error. El disseny de classes descrit amb UML ([28], Unified Modeling Language) es veu a la figura 6.1.

Com es pot veure al UML hi han tres classes principals *Function*, *Error* i *Method*. L'algorisme iteratiu representat per un objecte *method* de la classe *Method* resol una equació no lineal definida per un objecte *function* de la classe *Function* determinant el càlcul de l'error en cada pas d'iteració definit per un objecte *error* de la classe *Error*.

Es disposa de mètodes iteratius d'un i de dos passos. Els mètodes d'un pas poden formar els mètodes de dos passos mitjançant la congelació de l'operador de diferències dividides, *FrozenMethod*, o la pseudo-composició de dos mètodes, *PCMethod*. És per aquest motiu que representem els mètodes de dos passos *2Step* com l'agregació de dos objectes de la classe *1Step*.

És a dir, mitjançant un objecte *pcMethod* de la classe *PCMethod* es poden crear els següents mètodes:

- SKMethod (3.1.1)
- SStMethod (3.1.2)

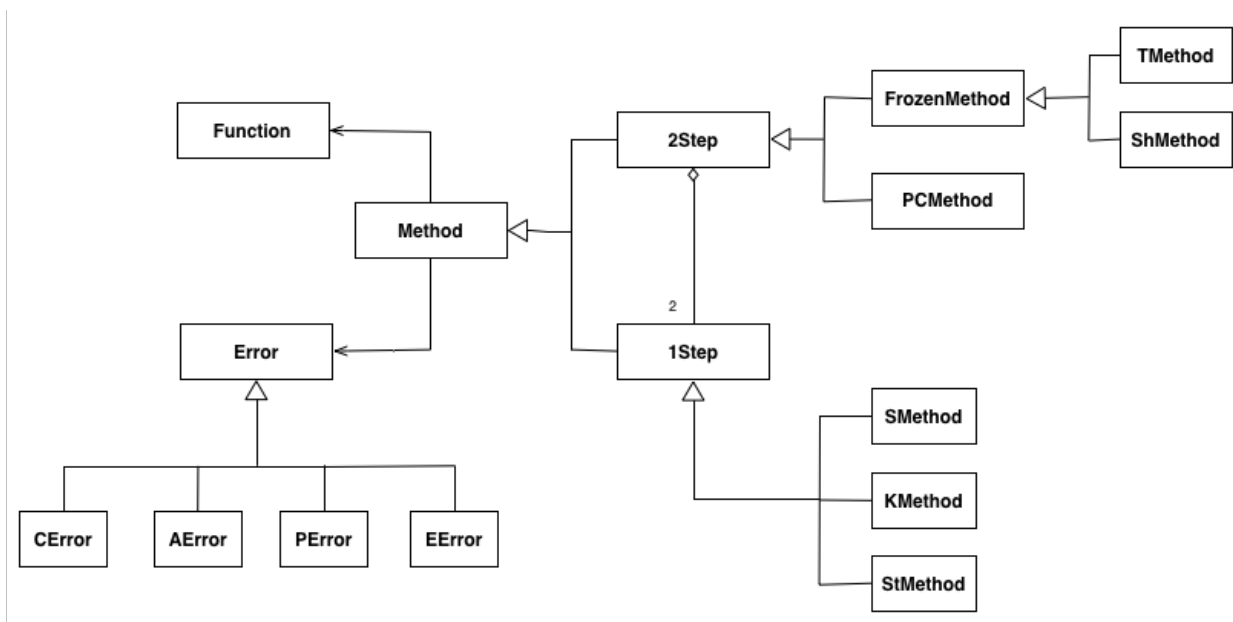


Figura 6.1: UML general de les classes.

- KSMMethod (3.1.3)
- KStMethod (3.1.4)
- StSMMethod (3.1.5)
- StKMethod (3.1.6)
- SSMMethod (3.1.10)
- KKMMethod (3.1.11)
- StStMethod (3.1.12)

Mitjançant un objecte de *frozenMethod* de la classe *FrozenMethod* es poden construir els següents esquemes iteratius mitjançant la congelació de l'operador.

- FSMMethod (3.1.7)
- FKMethod (3.1.8)
- FStMethod (3.1.9)

Les classes *TMethod* (3.1.14) i *ShMethod* (3.1.13) implementen mètodes de dos passos congelats.



Un objecte *error* pot ser de tipus *CError* (2.5.1), *AError* (2.5.2), *EError* (2.5.3) i *PError* (2.5.4). El càlcul de l'error en cada pas d'iteració estarà condicionat pel tipus del objecte que el defineix.

L'objecte *function* de la classe *Function* permet implementar qualsevol equació. A tal fi la figura 6.2 mostra la implementació realitzada de la classe. L'objecte *function* té com atribut un punter a la funció que implementa i un mètode *eval* que avalua la funció en un punt.

Function
void (*fun)(type&, type&); type alpha; int numProd, numQuoc, numFunc; double mu;
Function(void (*fcall)(type&,type&),int prec); void eval(type&, type&); double getCost(); void setCost(int c, int numProd, int numQuoc, int numFunc);

Figura 6.2: UML classe Function.

La classe *Error* està implementada como una classe virtual pura. Es veu a la figura 6.3. Les subclasses implementen les funcions però es fan les mateixes crides per a qualsevol tipus d'error.

Error
type en1,e1,e0; type p, pred; type cl, conv; type dn; int prec, digits;
Error(int prec, int digits); virtual int error(type& xn1, type& x1, type& x0)=0; virtual void criteria(type& rho)=0; virtual void localOrderConvergence()=0; virtual void decimals()=0; virtual void prediction(type& rho, double h)=0;

Figura 6.3: UML classe Error.

La classe *Method* té dos mètodes que cal destacar. La implementació de la classe es pot veure a la figura 6.4. Es defineix el mètode *iterate* con un mètode virtual pur que qualsevol subtipus de la classe *Method* haurà d'implementar. El mètode *solution* serà el mateix per a tots els subtipus. Amb aquesta combinació s'aconsegueix un mètode molt flexible que permetrà resoldre una equació no lineal amb qualsevol mètode implementat.

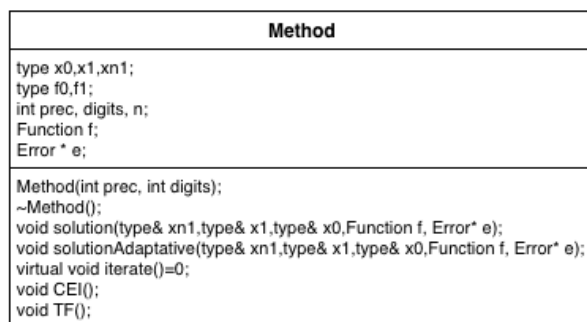


Figura 6.4: UML classe Method.

## 6.2 Algorisme genèric

L'algorisme genèric que implementa el mètode *solution* de la classe *Method* és pot veure a l'algorisme 1. Aquest algorisme empra l'aritmètica fixa. La versió del mètode *solution* amb aritmètica adaptativa es troba descrit a l'algorisme 2. Donades unes aproximacions inicials  $x_0$  i  $x_{-1}$ , el mètode  $m$  (2.2.1-3.1.14), construeix la successió d'iterats fins que l'iterat  $x_I$  aconsegueix el criteri de tall. Segons el paràmetre d'error el criteri canvia. Es poden veure les diferents alternatives a (2.6.4)-(2.6.7).

La principal diferència entre els algorismes 1 i 2 és el càlcul i la actualització de l'aritmètica que realitza el segon algorisme.

**Algorithm 1:** Algorisme genèric de resolució d'equacions no lineals a partir de mètodes iteratius amb aritmètica fixa.

**Data:** Aproximacions inicials  $x_0$ ,  $[x_{-1}]$ , Mètode  $m$ , precisió  $\eta$ , Ordre de convergència del mètode  $\rho$

**Result:** Iterat  $x_I$

Avaluar  $f(x_{-1})$  i  $f(x_0)$ ;

converge = false ;

**while not convergence do**

$x_{n+1}$  = iterar( $x_n, x_{n-1}$ );

$e_{n+1}$  = error( $x_{n+1}$ );

$d_{n+1}$  = decimals( $e_{n+1}$ );

**if**  $d_{n+1} < \eta$  **and**  $\rho d_{n+1} > \eta$  **then**

        converge = true ;

**else**

        Avaluar  $f(x_{n+1})$ ;

        increment n;

**end**

**end**

**return**  $x_{n+1}$ ;

**Algorithm 2:** Algorisme genèric de resolució d'equacions no lineals a partir de mètodes iteratius amb aritmètica adaptativa.

**Data:** Aproximacions inicials  $x_0$ ,  $[x_{-1}]$ , mètode  $m$ , precisió  $\eta$ , Ordre de convergència del mètode  $\rho$

**Result:** Iterat  $x_I$

Avaluar  $f(x_{-1})$  i  $f(x_0)$ ;

converge = false ;

**while not convergence do**

$x_{n+1}$  = iterar( $x_n, x_{n-1}$ );

$e_{n+1}$  = error( $x_{n+1}$ );

$d_{n+1}$  = decimals( $e_{n+1}$ );

**if**  $d_{n+1} < \eta$  **and**  $\rho d_{n+1} > \eta$  **then**

        converge = true ;

**else**

        incrementar precisió segons la predicció;

        Avaluar  $f(x_{n+1})$ ;

        increment n;

**end**

**end**

**return**  $x_{n+1}$ ;

## Capítol 7

# Planificació del projecte

### 7.1 Planificació temporal

El projecte consta de tres fases principals:

1. Planificació del projecte. Fita inicial.
  - (a) Abast i objectius.
  - (b) Planificació temporal
  - (c) Gestió econòmica i sostenibilitat
  - (d) Estat de l'art
2. Desenvolupament del projecte
  - (a) Cerca d'informació sobre els esquemes iteratius.
  - (b) Estudi, demostració i classificació dels mètodes iteratius a partir dels ordres de convergència local.
  - (c) Implementació dels mètodes iteratius amb aritmètica fixa i adaptativa, i càlcul dels paràmetres d'error.
  - (d) Proves/Validació dels resultats.
  - (e) Estudi de la convergència dels mètodes més eficients.
  - (f) Estudi del radi de convergència dels esquemes iteratius més eficients.
  - (g) Implementació dels mètodes amb altres programaris per tal de verificar les eficiències dels mètodes iteratius.
3. Fita final i conclusions.

### 7.1.1 Tasques

La primera fase del projecte, fita inicial, consisteix en la planificació de l'abast i objectius del projecte com també formalització de les tasques, la seva planificació temporal i el cost econòmic del treball de final de grau.

Aquesta primera fase consta de quatre tasques:

1. Definició de l'abast del projecte i objectius.
2. Planificació temporal de les fases i tasques del projecte.
3. Gestió econòmica i sostenibilitat.
4. Estat de l'art i bibliografia.

Es desenvoluparà en el període màxim de quatre setmanes.

Les dues següents fases del projecte, es realitzen per tant en el temps restant. Aproximadament tres mesos.

La fase de desenvolupament és la més extensa. Les tasques generals que es realitzen són les següents:

1. Estudi dels mètodes iteratius clàssics que utilitzen l'operador en diferències dividides.
2. Construcció de mètodes de dos passos a partir dels mètodes clàssics estudiats.
3. Demostració teòrica dels ordres de convergència dels esquemes iteratius.
4. Classificació dels algorismes iteratius a partir del seu ordre de convergència i cost computacional de forma teòrica.
5. Estudi dels diferents paràmetres d'error.
6. Disseny i avaluació del millor model de programació per la implementació del projecte en el llenguatge de programació C++.
7. Implementació, codificació del projecte amb aritmètica fixa amb multiprecisió (4096 dígitos).
8. Elecció i implementació de les funcions mostra.
9. Determinació dels intervals de convergència per a cada funció mostra tenint en compte la convergència i el número d'iteracions òptim.
10. Obtenció de resultats dels mètodes teòricament més eficients per a totes les funcions mostra presentades amb els intervals ja determinats.
11. Estudi teòric dels radis de convergència dels diferents esquemes iteratius més eficients.

12. Implementació dels mètodes pràctics d'estudi d'interval de convergència.
13. Adaptació de la implementació i codificació del projecte amb aritmètica adaptativa.
14. Extracció de resultats que demostrin la millora de rendiment i sostenibilitat del projecte en l'aplicació de l'aritmètica adaptativa.
15. Implementació dels mètodes iteratius amb altres programaris i estudi comparatiu dels temps d'execució.
16. Escriptura de la memòria i annex.
17. Conclusions del projecte.

En aquesta fase de desenvolupament es fa la codificació de tot el projecte.

Finalment, la darrera fase representa el tancament del projecte amb la seva defensa i la presentació de la memòria escrita.

És en aquesta última fase on es poden valorar els objectius assolits de forma continuada i concloure amb resultats i observacions el treball dels mesos anteriors.

### **7.1.2 Requeriments i anàlisi de dependències**

Els requeriments per a aquest projecte vénen determinats per les tasques a realitzar que s'han detallat a la secció 7.1.1.

L'estudi dels mètodes iteratius clàssics que utilitzen l'operador en diferències dividides requereix

- La comprensió del llenguatge i l'estil matemàtic utilitzat en la investigació sobre temes de càlcul numèric.
- La comprensió de les tècniques de presentació dels treballs d'investigació.

La tasca de construcció de mètodes de dos passos a partir dels mètodes clàssics necessita l'estudi anterior dels mètodes iteratius.

La demostració teòrica dels ordres de convergència dona lloc a la classificació dels algorismes. D'una banda la demostració teòrica dels ordres de convergència dels esquemes iteratius implica l'estudi de les tècniques de demostració de les equacions d'error dels mètodes iteratius. D'altra banda la classificació dels algorismes es farà a partir dels ordres de convergència i els paràmetres de cost computacional; és a dir, es classificarà segons l'eficiència dels mètodes. No es podrà realitzar fins el coneixement d'aquests paràmetres i ordres.

La implementació o codificació necessita que les tasques següents hagin estat realitzades:

- Estudi i configuració de la llibreria MPFR.
- Estudi i comprensió dels diferents paràmetres d'error i del càlcul de l'ordre de convergència local (ACLOC, CLOC, ECLOC i PCLOC).
- Anàlisi del model de programació i disseny de l'estructura de classes per a la codificació en C++.

Per poder fer l'extracció de resultats s'ha hagut de fer

- La codificació dels mètodes iteratius.
- L'elecció i la implementació de les funcions mostra.
- La determinació dels intervals de convergència òptims per a cada funció.

La implementació de l'aritmètica adaptativa es fa una vegada s'hagi realitzat amb l'aritmètica fixa y de manera que els paràmetres de convergència local donin els resultats correctes.

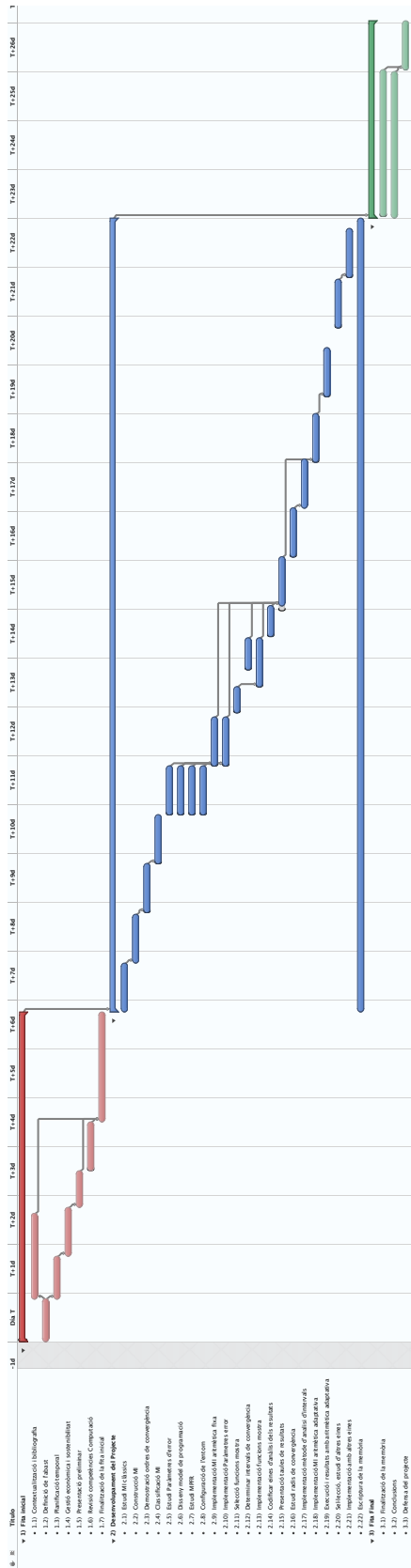
L'estudi teòric dels radis de convergència dels mètodes es pot realitzar en qualsevol moment posterior a la classificació dels mètodes iteratius més eficients i en tant en que hi hagi una comprensió de les tècniques necessàries. L'estudi pràctic es pot realitzar després de la codificació del cos principal del programa. Es fa en Maple [23].

Per fer la implementació dels mètodes iteratius amb altres programaris es considera una cerca d'informació per tal de considerar només les eines actuals més interessants i es realitza l'estudi dels programaris per tal de fer una implementació eficient.

L'escriptura de la memòria es fa de forma paral·lela a la realització de la resta de tasques del projecte. És necessari el coneixement tècnic de  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . Es pot descompondre en diferents tasques que es poden escriure segons avança el projecte.



Figura 7.1: Diagrama de Gantt



## 7.2 Planificació econòmica

### 7.2.1 Identificació de costos

El treball realitzat en aquest projecte de final de grau és un treball de recerca matemàtica.

El cost econòmic del treball realitzat consta de la suma de les hores invertides pels membres que han col·laborat en la realització del projecte de final de grau i els recursos hardware i software utilitzats pel seu desenvolupament.

A causa de l'àmbit del projecte, els rols que hi participen són:

- Investigador novell
- Programador
- Director del projecte
- Col·laboradors del grup d'investigació.

Cada tasca del diagrama de Gantt té associat un o més rols que la desenvolupen. El comput final d'hores de treball de cada rol es determina per tant a partir de la suma del cost temporal de totes les tasques d'un mateix rol.

A la taula es detallen les hores dedicades per part de cada rol i el cost associat.

### 7.2.2 Estimació de costos

El cost final del projecte és la suma dels costos de recursos humans, hardware i software descrits a continuació.

#### Recursos humans

Rol	Hores	Preu/h	Cost
Director del Projecte	100	40,00€	4.000,00€
Investigador novell	600	25,00€	15.000,00€
Programador	280	25,00€	7.000,00€
Col·laboradors d'investigació	10	40,00€	400,00€
Total	890		26.400,00€

Taula 7.1: Cost econòmic del projecte.

### Recursos de hardware

MacBook Pro amb processador 2.7 GHz Intel Core i7 i una memòria de 16 GB 1600 MHz amb un sistema operatiu MacOS. El cost actual és 2.529,00€ amb una vida útil d'aproximadament 5 anys. L'amortització és de 168,60€.

### Recursos de software

De tot el programari utilitzat només la llicència de *Maple* té repercussió en el cost econòmic del projecte i és de 100,00€. La resta de programari (*Xcode*, *OSX Yosemite*, *Python*, *MPFR*, *GMP*, *ARPREC*, *BigFloat*) és de cost 0,00€.

En definitiva, el cost final és de 26.668,60€.

## 7.3 Sostenibilitat i compromís social

La matriu de sostenibilitat és troba a la taula 7.2.

Sostenible	Econòmica	Social	Ambiental
Planificació	Viabilitat econòmica 8	Millora en la qualitat de vida 8	Anàlisi de recursos 8
Resultats	Cost final vs. previsió 10	Impacte en l'entorn social 8	Consum de recursos 8
Riscos	Adaptació a canvis d'escenari 9	Dany social 9	Dany mediambiental 9
Valoració total	9	8	8

Taula 7.2: Matriu de sostenibilitat.



## Capítol 8

# Conclusions

Aquest treball ha estat focalitzat en la resolució d'equacions no lineals amb l'operador diferències dividides. A partir dels mètodes iteratius clàssics, s'ha fet la construcció de nous esquemes iteratius fent servir la pseudo-composició i la congelació de l'operador amb l'objectiu de trobar algorismes amb un ordre de convergència superior i un cost computacional baix.

Com a resultat de l'estudi teòric dels ordres de convergència local i els costos computacionals de cada mètode iteratiu s'ha fet una classificació en termes d'eficiència dels algorismes iteratius presentats. Finalment, s'ha realitzat un estudi teòric dels radis de convergència dels quatre mètodes amb un índex d'eficiència computacional més gran.

Per obtenir els resultats numèrics s'ha fet servir un conjunt de funcions mostra variat amb uns costos computacionals associats que pretenen comprovar el que s'ha estudiat teòricament.

Com a resultats numèrics es presenta la resolució de les equacions no lineals mostra amb els mètodes iteratius més eficients fent servir una aritmètica fixa i una aritmètica adaptativa. Es comprova el comportament dels algorismes amb diferents aproximacions inicials.

El sistema de resolució implementat està pensat com un sistema flexible capaç de resoldre qualsevol equació no lineal fent servir qualsevol mètode iteratiu o combinació de mètodes emprant les tècniques de pseudo-composició o congelació. Qualsevol paràmetre d'error pot ser utilitzat per l'avaluació dels iterats.

Aquest sistema s'ha replicat amb tres llibreries diferents per tal de comparar els temps de resolució. En general la llibreria més eficient és MPFR tot i que BigFloat té un comportament molt similar i, en alguns casos, l'agilitat de python pot ser avantatge.

Es pot concloure que el mètode iteratiu més eficient per a la resolució de funcions escalars és el mètode de la Secant en termes d'index de cost computacional. Tot i així, el radi de convergència i les aproximacions inicials juguen un paper molt important en el comportament final de l'algorisme i varien en funció del mètode i de l'equació que es pretén resoldre. Sense cap dubte, l'aritmètica adaptativa és una gran millora respecte l'aritmètica fixa. Tant en termes de temps d'execució com també d'estalvi de recursos.

Com a feina futura s'hauria de considerar la generalització de les tècniques utilitzades a dimensió superior. Fer un estudi de la implementació en paral·lel dels mètodes iteratius i la construcció de nous esquemes a partir dels mètodes de Shakno i Traub.

# Bibliografia

- [1] A. Aitken. On Bernoulli's numerical solution of algebraic equations. *Proc. Roy. Soc. Edinburgh*, 46:289–305, 1927.
- [2] S. Amat and S. Busquier. On a Steffensen's type method and its behavior for semismooth equations. *Appl. Math. Comput.*, 177(2):819–823, 2006.
- [3] S. Amat and S. Busquier. A two-step Steffensen's method under modified convergence conditions. *J. Math. Anal. Appl.*, 324(2):1084–1092, 2006.
- [4] I. Argyros. On the secant method. *Publ. Math. Debrecen*, 43(3-4):223–238, 1993.
- [5] D. H. Bailey. High-precision computation and mathematical physics. *Lawrence Berkeley National Laboratory*, 2009.
- [6] D. H. Bailey, H. Yozo, X. S. Li, and B. Thompson. ARPREC: An arbitrary precision computation package. *Lawrence Berkeley National Laboratory*, 2002.
- [7] M. Dickinson. The BigFloat package high precision floating-point arithmetic <http://bigfloat.readthedocs.org/en/latest/#>.
- [8] J. Ezquerro, A. Grau, M. Grau-Sánchez, and M. Hernández. A new class of secant-like methods for solving nonlinear systems of equations. *Commun. Appl. Math. Comput. Sci.*, 9(2):201–213, 2014.
- [9] J. Ezquerro, A. Grau, M. Grau-Sánchez, M. Hernández, and M. Noguera. Analysing the efficiency of some modifications of the secant method. *Comput. Math. Appl.*, 64(6):2066–2073, 2012.
- [10] J. Ezquerro, M. Grau-Sánchez, M. Hernández-Verón, and M. Noguera. A traub type result for one-point iterative methods with memory. *Anal. Appl.*, 12(03):323–340, 2014.
- [11] J. Ezquerro, M. Grau-Sánchez, M. Hernández-Verón, and M. Noguera. A family of iterative methods that uses divided differences of first and second orders. *Numer. Algorithms*, (DOI:10.1007/S11075-015-9962-0), 2015.

- [12] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélicier, and P. Zimmermann. MPFR: A multiple-precision binary floating-point library with correct rounding. *ACM Trans. Math. Softw. (TOMS)*, 33(2):13, 2007.
- [13] M. Grau-Sánchez and J. L. Diaz-Barrero. Note on improvements of order and efficiency in some iterative methods for solving nonlinear equations. *Appl. Math. Lett.*, 33:18–22, 2014.
- [14] M. Grau-Sánchez, À. Grau, and M. Noguera. Frozen divided difference scheme for solving systems of nonlinear equations. *J. Comput. Appl. Math.*, 235(6):1739–1743, 2011.
- [15] M. Grau-Sánchez, À. Grau, and M. Noguera. On the computational efficiency index and some iterative methods for solving systems of nonlinear equations. *J. Comput. Appl. Math.*, 236(6):1259–1266, 2011.
- [16] M. Grau-Sánchez and M. Noguera. A technique to choose the most efficient method between secant method and some variants. *Appl. Math. Comput.*, 218(11):6415–6426, 2012.
- [17] M. Grau-Sánchez, M. Noguera, and J. L. Diaz-Barrero. On the local convergence of a family of two-step iterative methods for solving nonlinear equations. *Comput. Math. Appl.*, 255:753–764, 2014.
- [18] M. Grau-Sánchez, M. Noguera, À. Grau, and J. R. Herrero. On new computational local orders of convergence. *Appl. Math. Lett.*, 25(12):2023–2030, 2012.
- [19] M. Grau-Sánchez, M. Noguera, and J. M. Gutiérrez. Frozen iterative methods using divided differences à la schmidt–schwetlick. *J. Optimiz. Theory Appl.*, 160(3):931–948, 2014.
- [20] M. Grau-Sánchez, A. Grau, and J. L. Diaz-Barrero. On computational order of convergence of some multi-precision solvers of nonlinear systems of equations. *Octagon Math. Magazine*, 21:7569–7592, 2014.
- [21] V. Kurchatov. On a method of linear interpolation for the solution of functional equations. *Dokl. Akad. Nauk SSSR*, 198(3):524–526, 1971.
- [22] M. N. M. Grau-Sánchez and J. Gutiérrez. On some computacional  $R$ -orders of convergence. *Appl. Math. Lett.*, 23:472–478, 2010.
- [23] V. Maple. Waterloo maple software. *University of Waterloo, Version 16*, 5, 1994.
- [24] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Academic Press New York, 1970.
- [25] A. Ostrowski. *Solutions of Equations and System of Equations*. Academic Press New York, 1960.



- [26] M. S. Petkovic. Remarks on on a general class of multipoint root-finding methods of high computational efficiency. *SIAM J. Numer. Anal.*, 49(3):1317–1319, 2011.
- [27] F. Potra and V. Pták. *Nondiscrete induction and iterative methods*. Pitman Publishing Limited, Boston, 1984.
- [28] J. Rumbaugh, I. Jacobson, and G. Booch. *Unified Modeling Language Reference Manual, The*. Pearson Higher Education, 2004.
- [29] S. Shakhno. On a Kurchatov’s method of linear interpolation for solving nonlinear equations. *Proc. Appl. Math. Mech.*, 4(1):650–651, 2004.
- [30] J. Steffensen. Remarks on iteration. *Scand. Act. J.*, 1933(1):64–72, 1933.
- [31] S. W. T. J. McDougall. A simple modification of newtons method to achieve convergence of order  $1 + \sqrt{2}$ . *Appl. Math. Lett.*, 29:20–25, 2014.
- [32] L. Tornheim. Convergence of multipoint iterative methods. *J. ACM*, 11(2):210–220, 1964.
- [33] J. F. Traub. *Iterative methods for the solution of equations*. Prentice-Hall, Englewood Cliffs, New Jersey, 1964.
- [34] D. Wall. The order of an iteration formula. *Math. Tables Aids Comput.*, 10(55):167–168, 1956.
- [35] S. Weerakoon and T. Fernando. A variant of Newton’s method with accelerated third-order convergence. *Appl. Math. Lett.*, 13(8):87–93, 2000.



## Apèndix A

# Taules complementàries dels resultats numèrics

Les taules que componen els resultats presentats en el treball es poden consultar al document adjunt *Mètodes per a la resolució d'equacions no lineals amb l'operador diferències dividides - Apèndix A*.

### A.1 Aritmètica Fixa (AF)

Per aritmètica fixa es presenten per als mètodes de la Secant ( $\phi_1$ ), Kurxatov ( $\phi_2$ ), Shakno ( $\phi_3$ ) i Frozen Steffensen ( $\phi_4$ ), una taula per funció  $f_i(x)$ ,  $1 \leq i \leq 10$ , on es comparen els dígitos correctes,  $D_n$ ,  $\hat{D}_n$ ,  $\tilde{D}_n$  i  $\check{D}_n$ , obtinguts pels paràmetres d'error  $e_n$ ,  $\hat{e}_n$ ,  $\tilde{e}_n$  i  $\check{e}_n$ .

### A.2 Aritmètica Adaptativa (AA)

Per aritmètica adaptativa es presenten dos tipus de taules per a cada mètode  $\phi_j$ ,  $1 \leq j \leq 4$ , i funció  $f_i(x)$ ,  $1 \leq i \leq 10$ . D'una banda les taules on es comprova el  $\%t_n$  de temps acumulat per iteració. D'altra banda la comparativa dels diferents paràmetres de predicció  $m_{n+1}$ ,  $\hat{m}_{n+1}$ ,  $\tilde{m}_{n+1}$  i  $\check{m}_{n+1}$ . El primer tipus de taula es presenta al costat de la mateixa execució per a l'aritmètica fixa.

### A.3 Comparativa entre AA i AF

En la comparativa de les dues aritmètiques es presenta una taula entre els temps d'execució amb aritmètica fixa (AF) i adaptativa (AA) per a cada mètode  $\phi_j$ ,  $1 \leq j \leq 4$ , i funció  $f_i(x)$ ,  $1 \leq i \leq 10$ . Les taules presentades banda a banda en l'apartat d'aritmètica adaptativa també serveixen per a la comparativa.



## Apèndix B

### Codi

#### B.1 Sistema de resolució d'equacions no lineals

El sistema de resolució d'equacions no lineals, que s'ha anomenat MNUM, s'ha desenvolupat sota tres programaris diferents. La versió més completa del sistema de resolució que admet tan aritmètica fixa com adaptativa és el realitzat amb MPFR i C++. Les altres dues versions només admeten aritmètic fixa. La versió que utilitza el mòdul BigFloat està desenvolupada amb Python. La que fa servir ARPREC està desenvolupada en C++.

A continuació es veu l'estructura de carpetes del contingut adjunt.

```
├── Tablas
└── Codi
    ├── Eficiencias
    ├── Intervals
    ├── KLM
    ├── src
    │   ├── MPFR
    │   │   ├── MNUMmpfr
    │   │   └── ExempleAA
    │   ├── ARPREC
    │   │   └── MNUMArprec
    │   ├── BigFloat
    │   │   └── MNUMBigFloat
```

##### B.1.1 MPFR

Tot el codi del sistema de resolució MNUM desenvolupat amb MPFR es troba a *src/MPFR/MNUMMpfr*.

### B.1.2 ARPREC

Tot el codi del sistema de resolució MNUM desenvolupat amb MPFR es troba a *src/ARPREC/MNUMArprec*.

### B.1.3 BigFloat

Tot el codi del sistema de resolució MNUM desenvolupat amb MPFR es troba a *src/BigFloat/MNUMBigFloat*.

## B.2 Eficiències

L'estudi de les eficiències s'ha realitzat amb un programa de maple. Es troba a l'arxiu *Eficiencies-dim.mws*

## B.3 Radis

L'estudi teòric dels radis de convergència s'ha realitzat mitjançant el mòdul *sympy* de Python. L'script es troba a *klm.py*.

## B.4 Aproximacions inicials i número d'iteracions

L'estudi pràctic de les aproximacions inicials es realitza amb el sistema de resolució original de MPFR. El codi per realitzar l'estudi es troba a *Codi/intervals/interval.cpp*.

